CSCI 1470/2470
Spring 2024

Ritambhara Singh

March 18, 2024
Monday

Deep Learning

Interpretability

ChatGPT prompt “minimalist landscape painting of a deep underwater scene with a blue tang fish in the bottom right corner”

Machine Learning and Interpretability

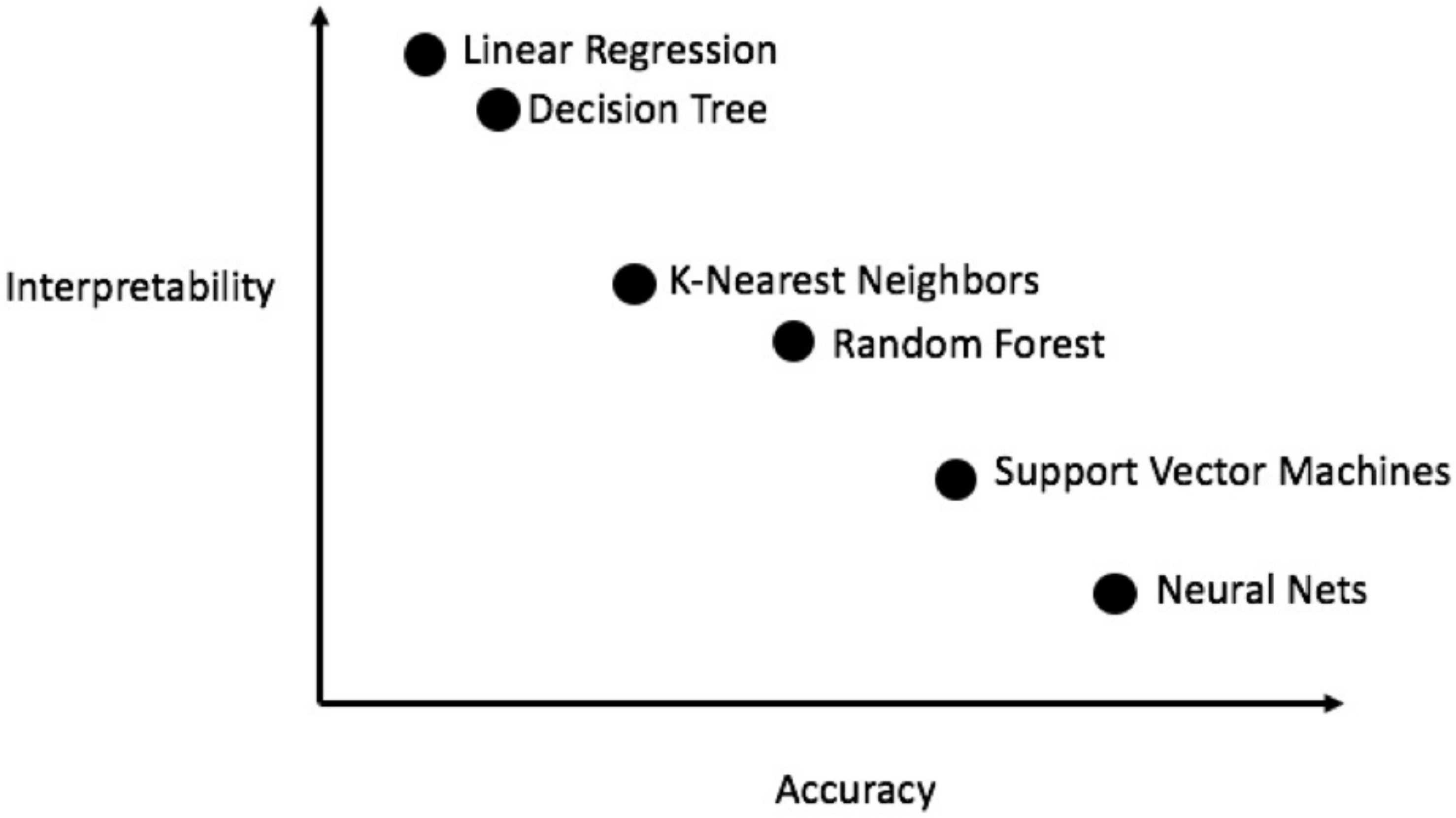
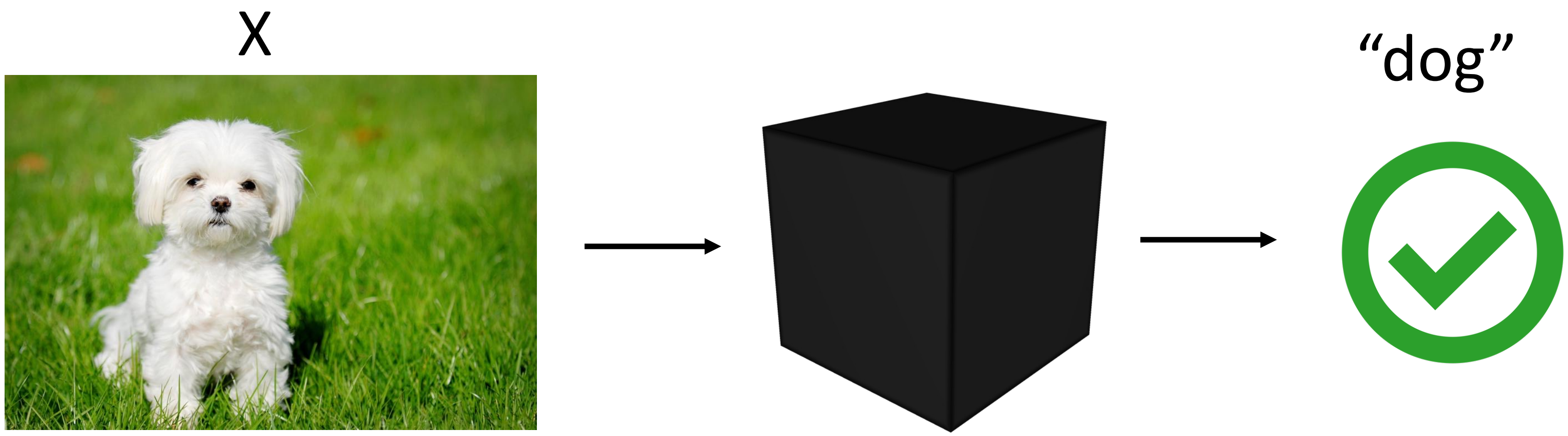


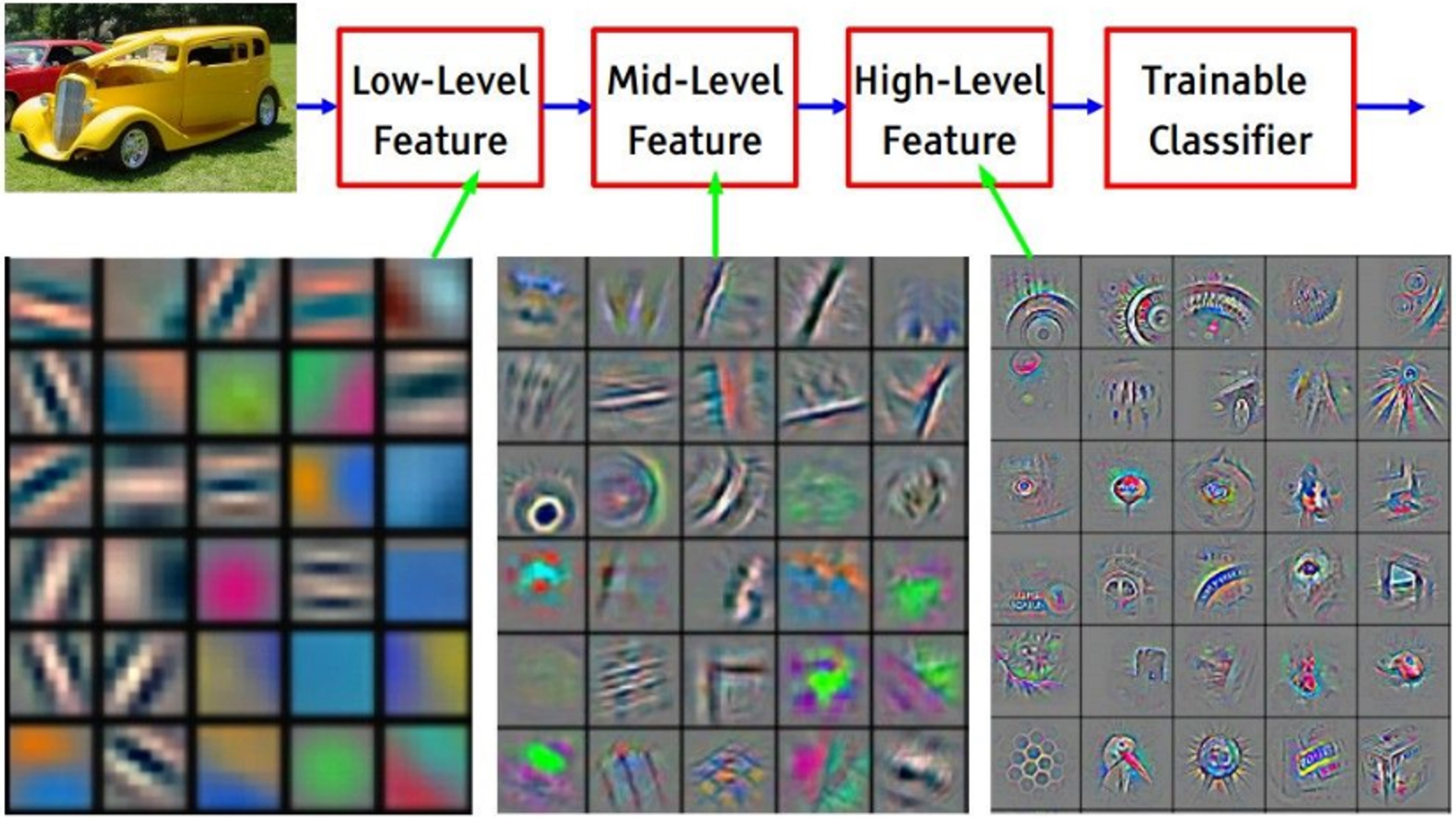
Image courtesy: https://medium.com/ansaro-blog/interpreting-machine-learning-models-1234d735d6c9

Deep Nets are “Black Box” Models

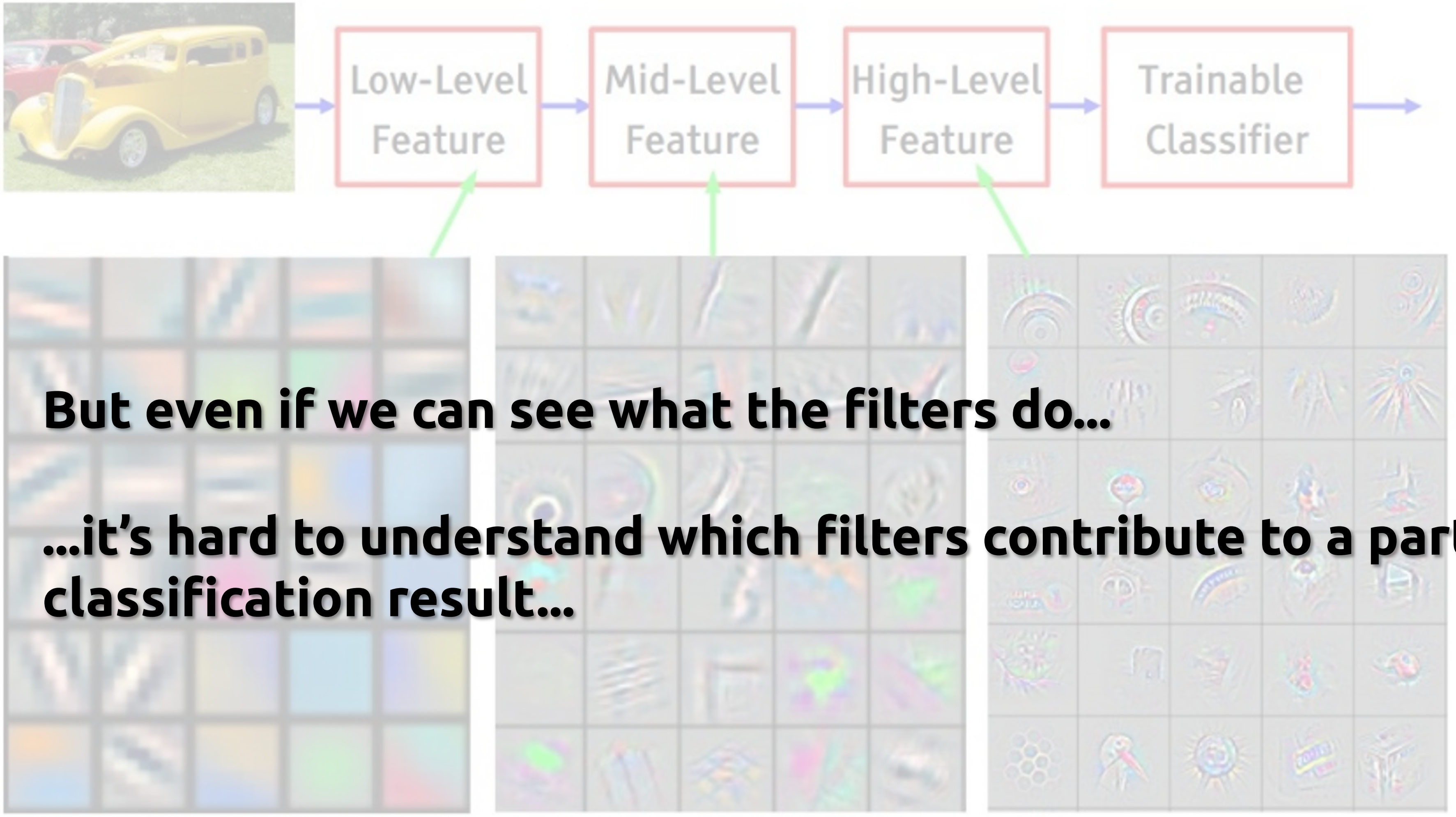
- What do the hidden layers of networks actually learn?
    - **Image recognition:** What do the many thousands of filters actually represent??
    - **Natural language processing**: What does the RNN hidden state actually store??



Example: What do CNN filters look like?



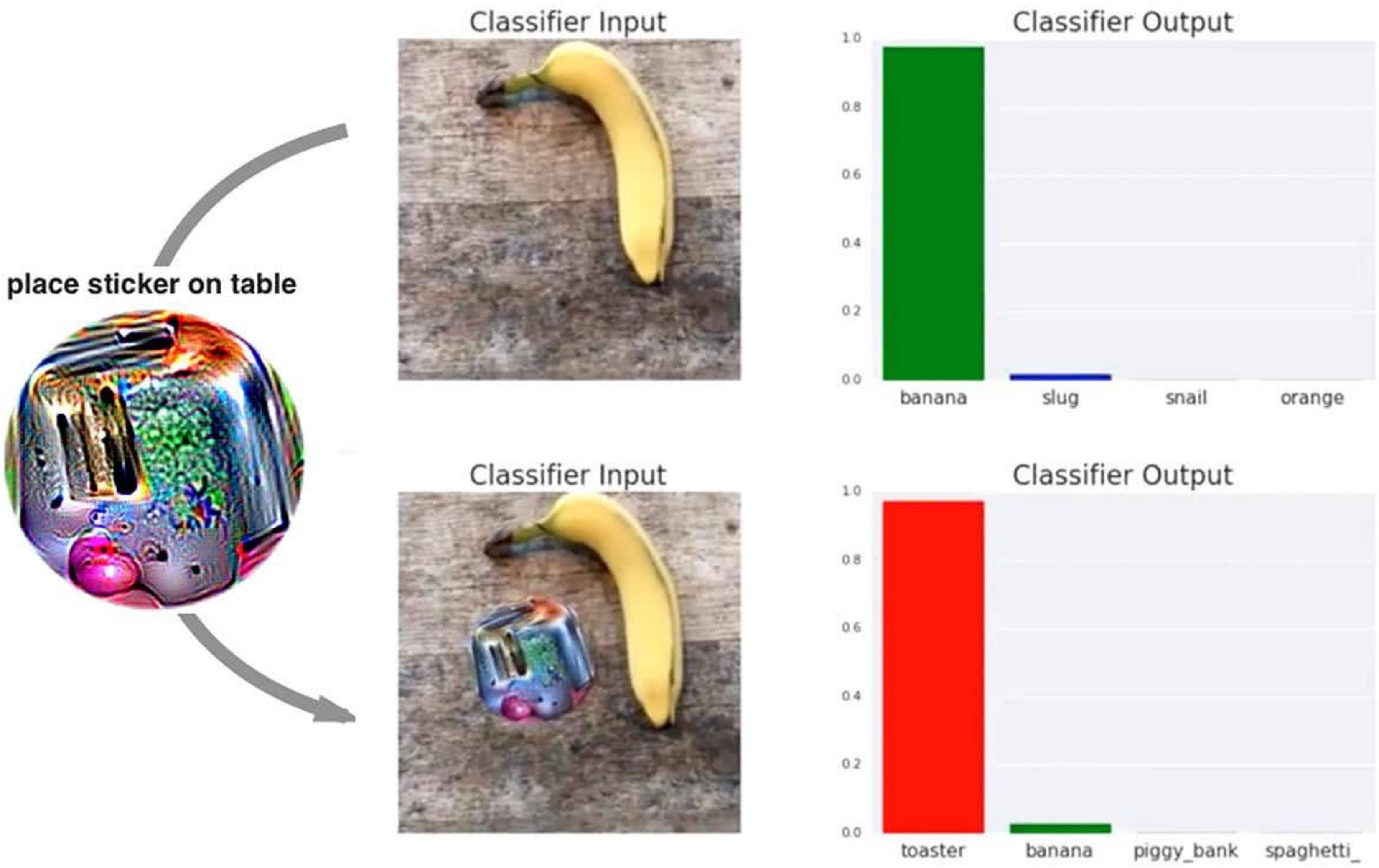
Example: What do CNN filters look like?



But even if we can see what the filters do…

…it's hard to understand which filters contribute to a particular classification result…

...which leads to situations like this:



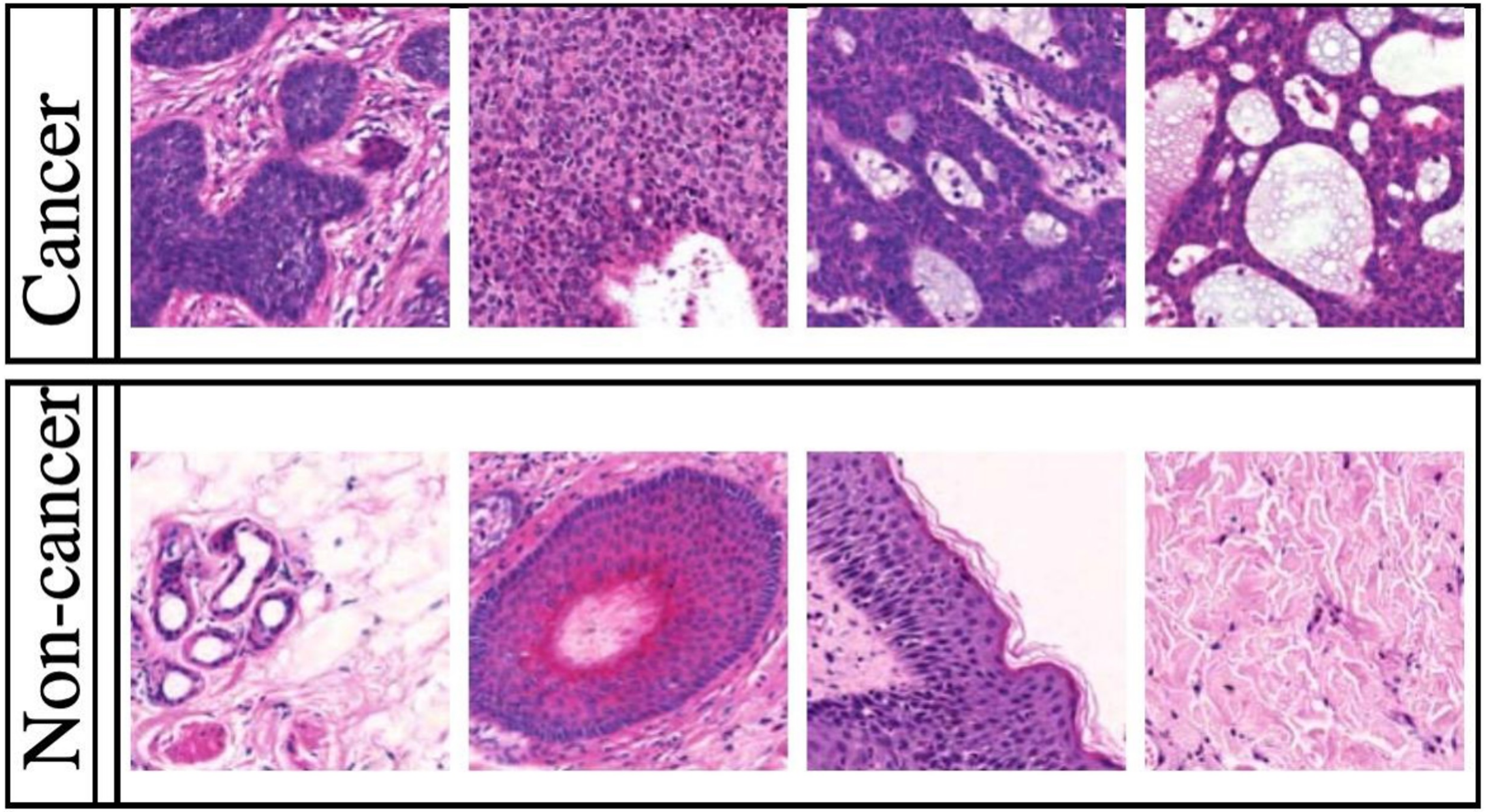
Deep Learning has an

interpretability problem

Maybe not such a big deal if we’re just classifying breakfast foods…

…but what if the decision is
really important?

What if the decision is *really* important?



From Cruz-Roa et al., 2013

What if the decision is *really* important?

- How can a human (e.g. a doctor) trust that a network is making a sensible decision (e.g. a patient has the flu, and *not* strep, mono, or pneumonia)?

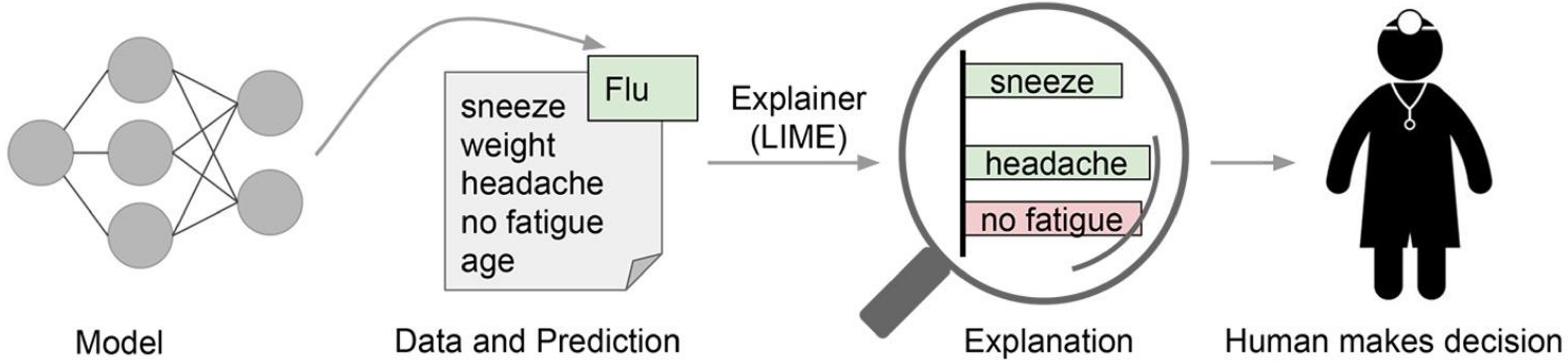


Figure 1. Explaining individual predictions to a human decision-maker. Source: Marco Tulio Ribeiro.

Model Interpretability

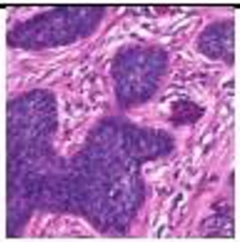
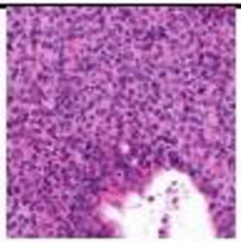
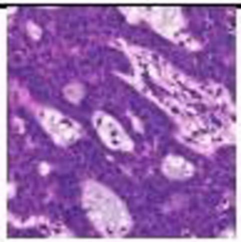
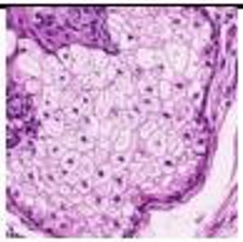
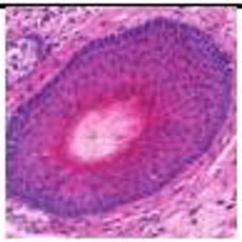
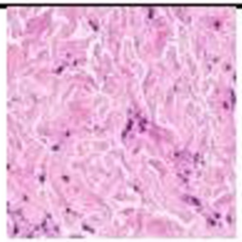
- Broadly, ***interpretability*** refers to ways of understanding/measuring how a model made a decision

- Can take the form of visualizations, summary statistics, metrics, ...

- A whole subfield of study: ***Interpretable DL/AI***

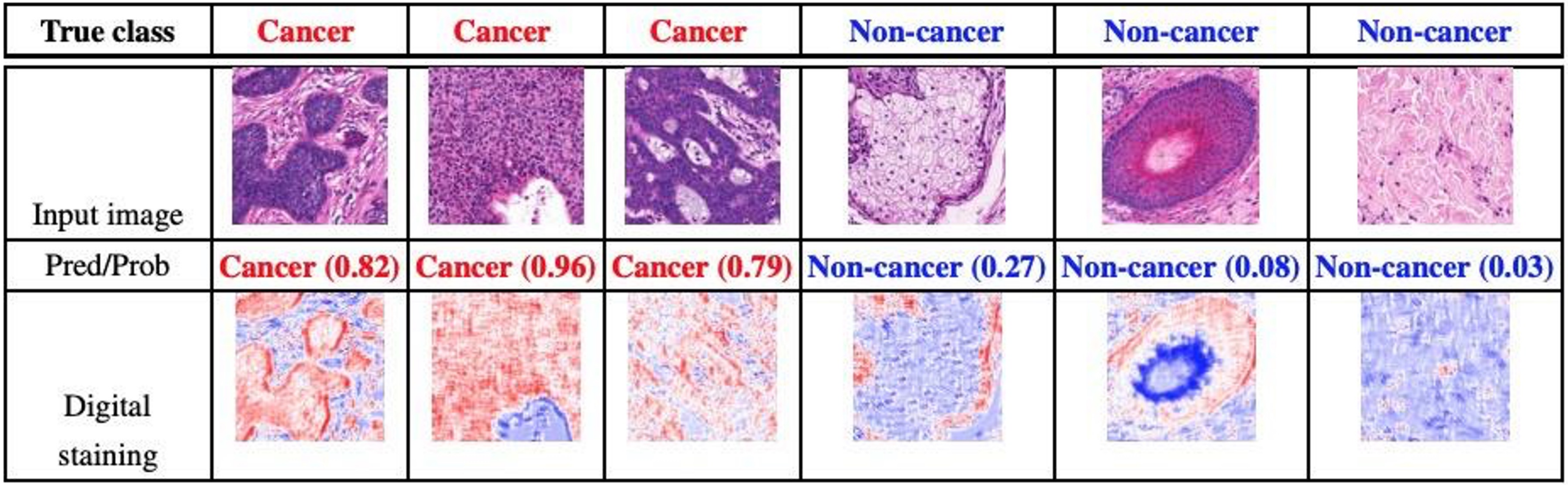
Making Cancer Predicting Interpretable

Simultaneously learn to predict which regions of the tissue are cancerous
(so a person can look and see if that makes sense)

True class	Cancer	Cancer	Cancer	Non-cancer	Non-cancer	Non-cancer
Input image						
Pred/Prob	Cancer (0.82)	Cancer (0.96)	Cancer (0.79)	Non-cancer (0.27)	Non-cancer (0.08)	Non-cancer (0.03)

Making Cancer Predicting Interpretable

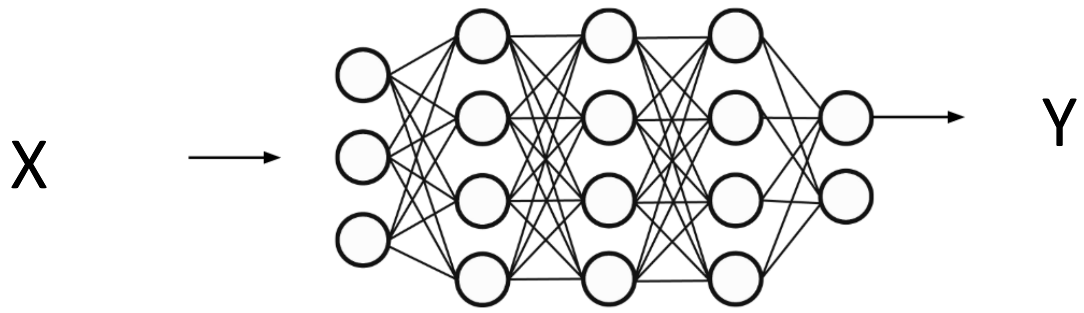
Simultaneously learn to predict which regions of the tissue are cancerous
(so a person can look and see if that makes sense)



True class	Cancer	Cancer	Cancer	Non-cancer	Non-cancer	Non-cancer
Input image						
Pred/Prob	Cancer (0.82)	Cancer (0.96)	Cancer (0.79)	Non-cancer (0.27)	Non-cancer (0.08)	Non-cancer (0.03)
Digital staining						

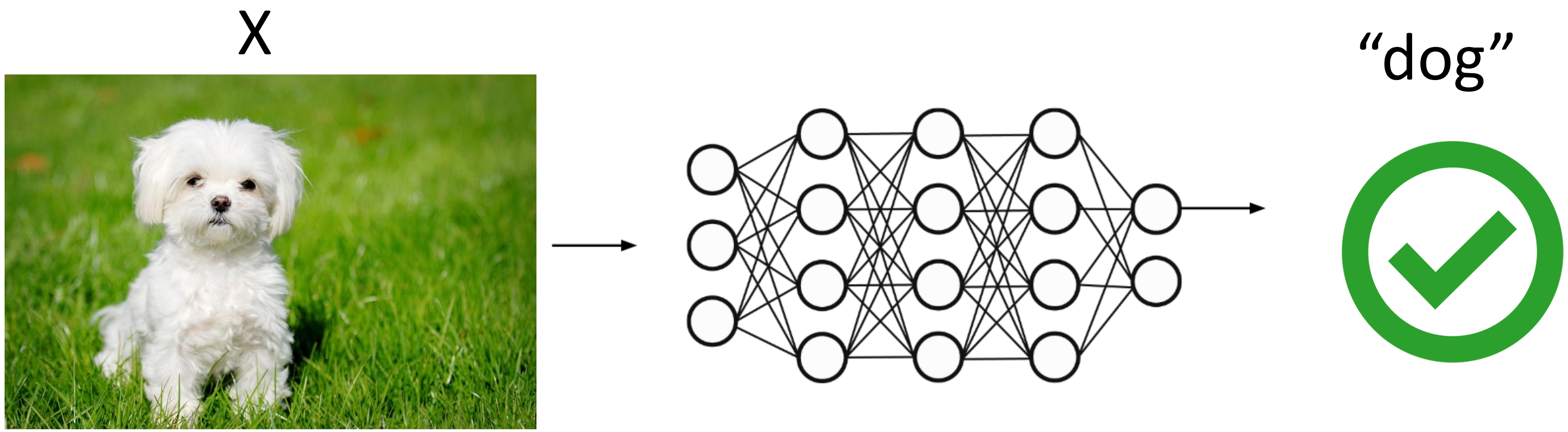
Red = cancerous tissue
Blue = healthy tissue

Task formulation



Which features in X are most important for the model prediction Y?

Task formulation

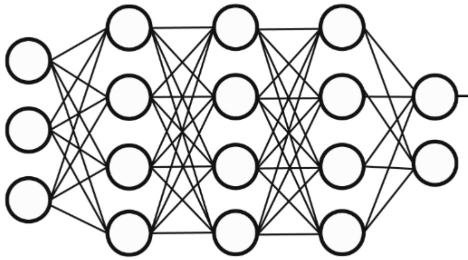


Which pixels are most important for classification?

Task formulation

X

I don't like the actors, but I really enjoyed this movie →



→ s_+

positive sentiment

Which words/characters are most important for classification?

Today’s goal – learn about interpretation in DL

(1) Model architecture based methods (CNNs and RNNs)

(2) Gradient-based methods

(3) Model agnostic methods

Note: categorization done loosely by me

Today's goal – learn about interpretation in DL

(1) Model architecture based methods (CNNs and RNNs)

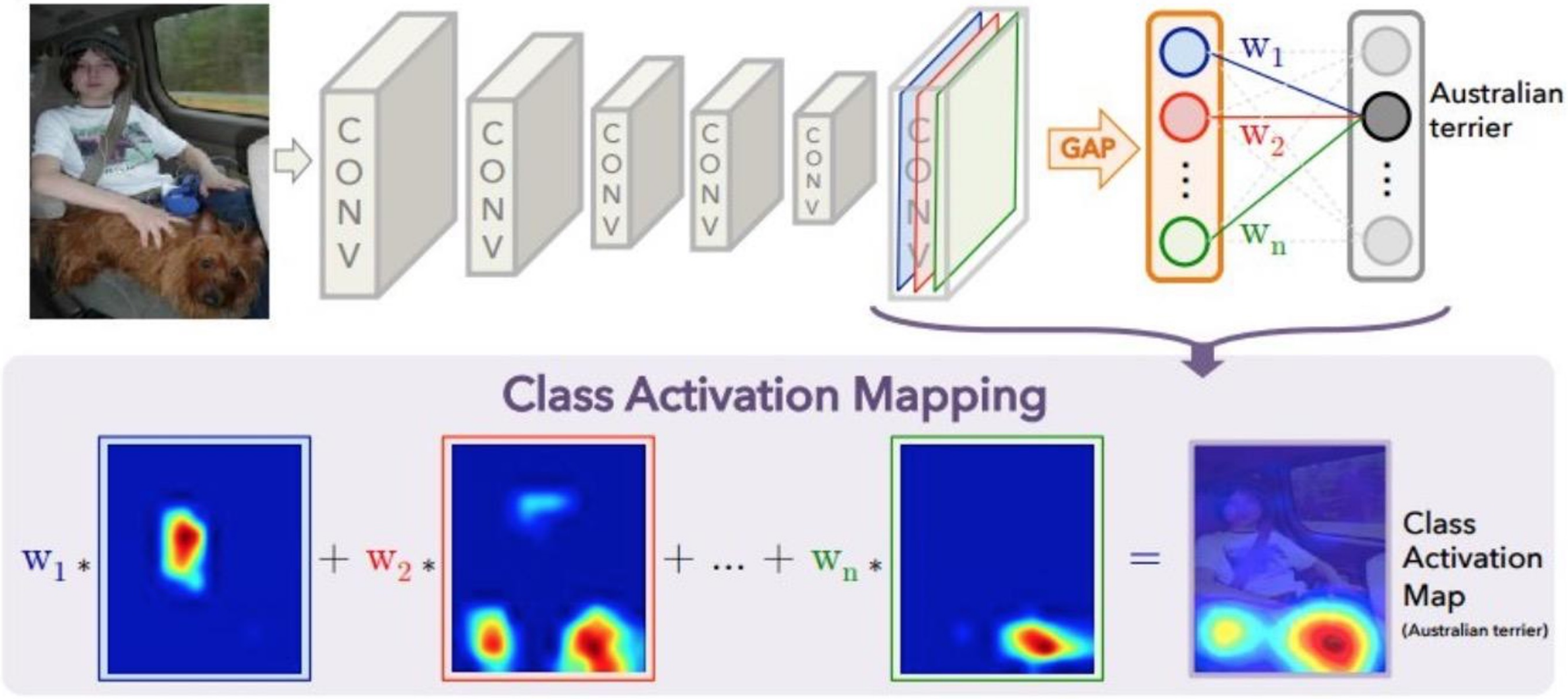
(2) Gradient-based methods

(3) Model agnostic methods

Identifying image regions that influence classification result

Global Average Pooling: Average all the pixels in the last feature map to produce a flat vector, then feed that through a linear layer to produce class logits

- A weighted sum of the last feature maps, according to the weights of the linear layer, localizes the region that leads to the classification

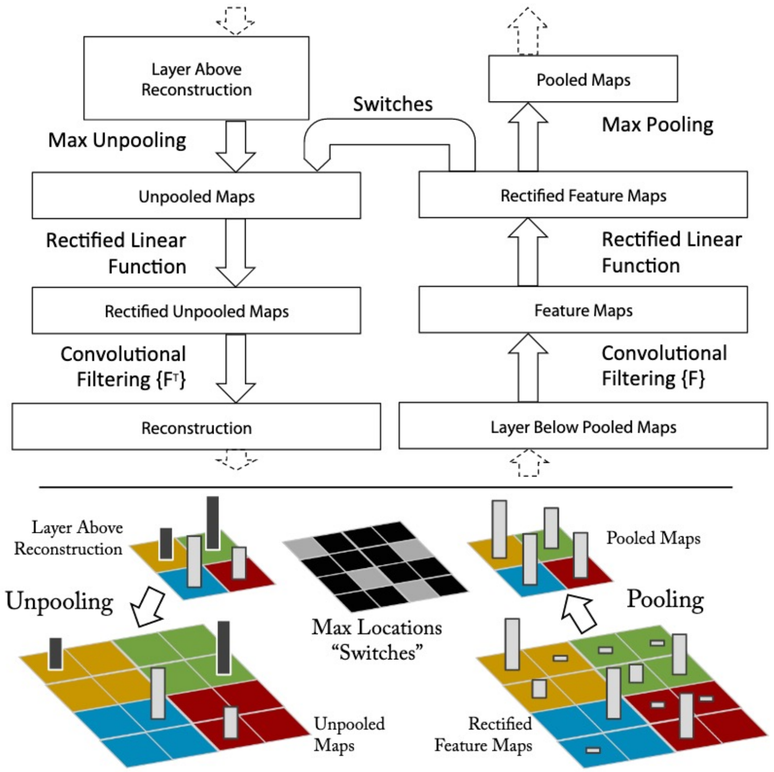


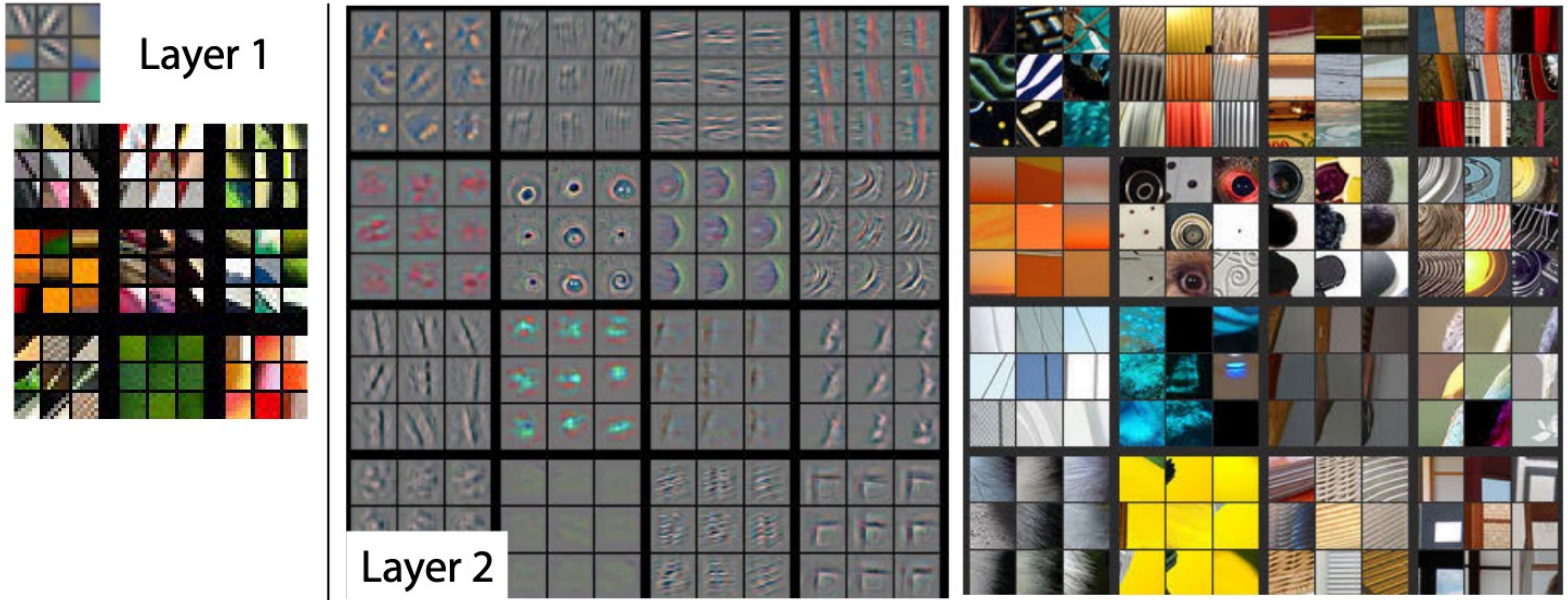
Deconvolution

Map filter activations back to the input pixel space, showing what input pattern originally caused a given activation in the feature maps.

Perform this mapping with a Deconvolutional Network (decovnet).

Decovnet: a convnet model that uses the same components (filtering, pooling) but in reverse, so instead of mapping pixels to features it does the opposite.



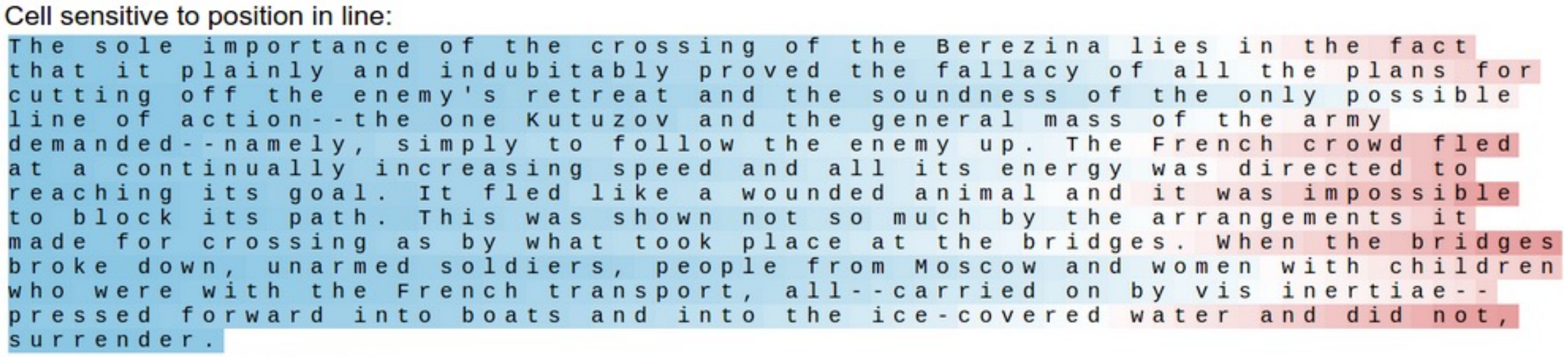


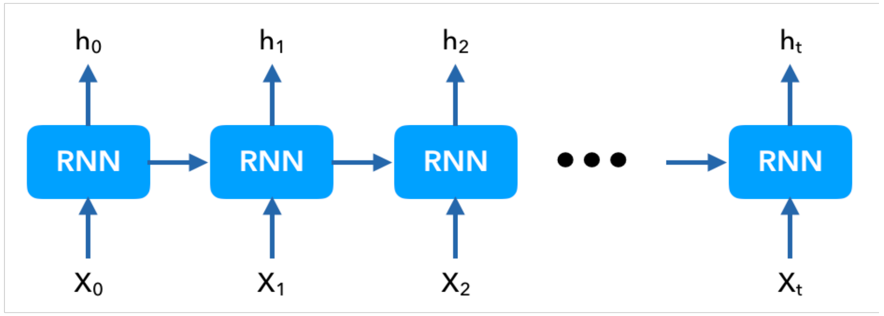
Visualizing and Understanding Convolutional Networks

Interpreting RNNs

Pick one entry (cell) of the hidden state, highlight characters that cause that cell to take on a high value

- This is a ***character-level language model***, not a word-level one



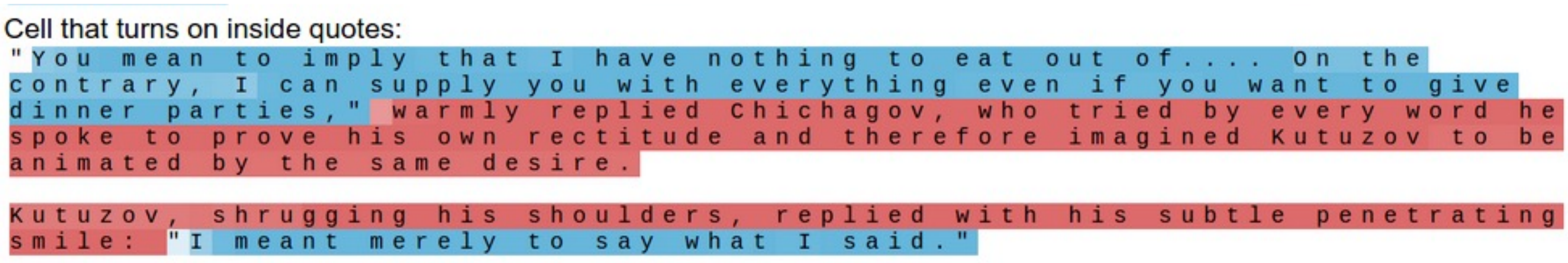


http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Interpreting RNNs

Pick one entry (cell) of the hidden state, highlight characters that cause that cell to take on a high value

- This is a ***character-level language model***, not a word-level one



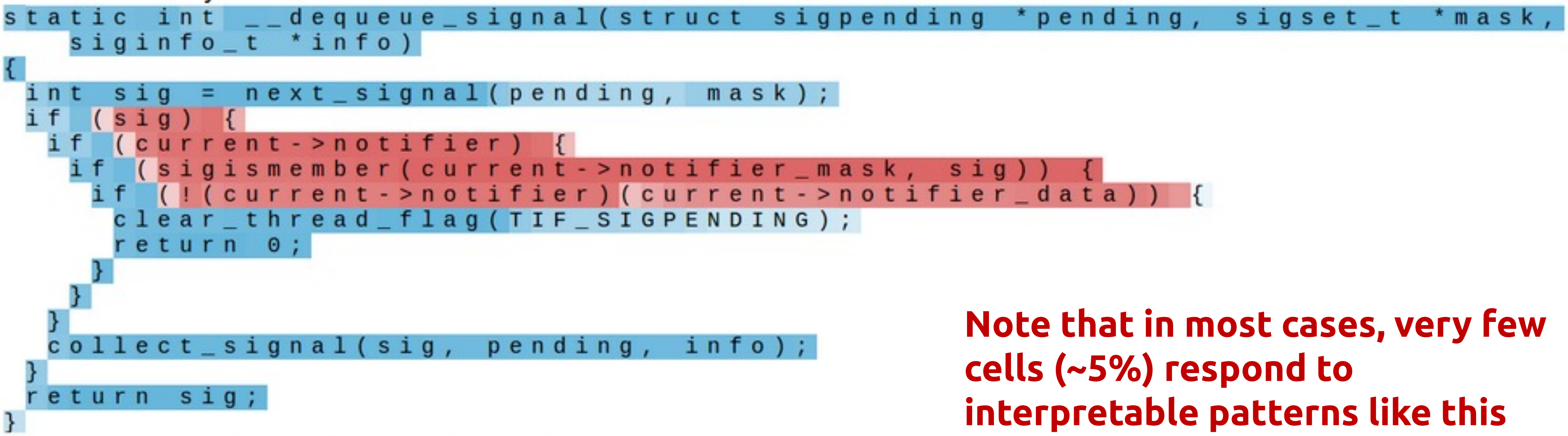
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Interpreting RNNs

Pick one entry (cell) of the hidden state, highlight characters that cause that cell to take on a high value

- This is a ***character-level language model***, not a word-level one

Cell that robustly activates inside if statements:



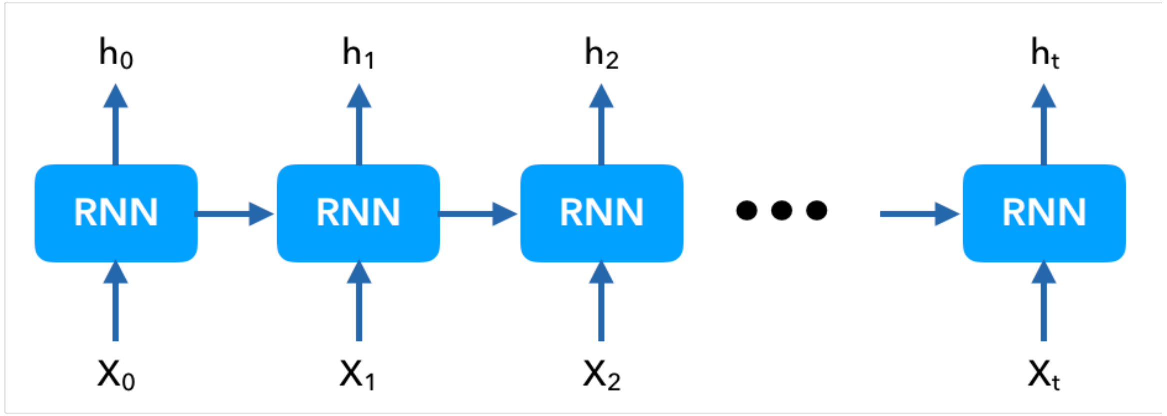
Note that in most cases, very few cells (~5%) respond to interpretable patterns like this

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Any questions?

Temporal output (RNNs)

Track the prediction of the RNN for one hidden unit at a time



Deep Motif Dashboard: Visualizing and Understanding Genomic Sequences Using Deep Neural Networks

Today’s goal – learn about interpretation in DL

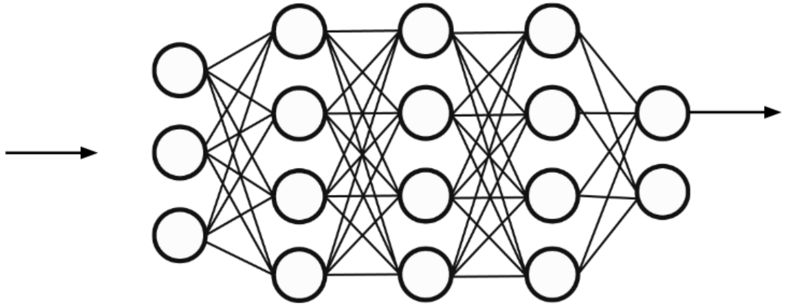
(1) Model architecture based methods

(2) Gradient-based methods

(3) Model agnostic methods

Saliency maps

X

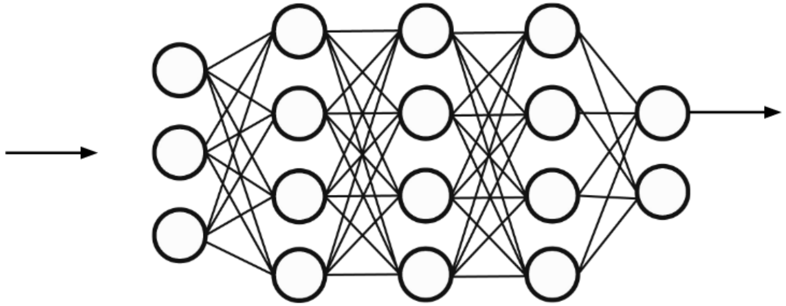


“dog”

Which pixels are most important for classification?

Saliency maps

X



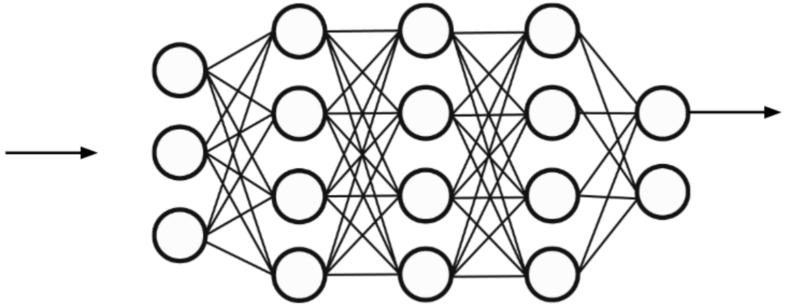
$S_+(X)$

$$S_+(X) \approx w^T X + b = \sum_{i=1}^{|X|} w_i x_i$$

[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps]

Saliency maps

X



$S_+(X)$

$$S_+(X) \approx w^T X + b = \sum_{i=1}^{|X|} w_i x_i$$

$$w = \left. \frac{\partial S_+}{\partial X} \right|_{X_0} = \text{“}\textit{saliency map}\text{”}$$

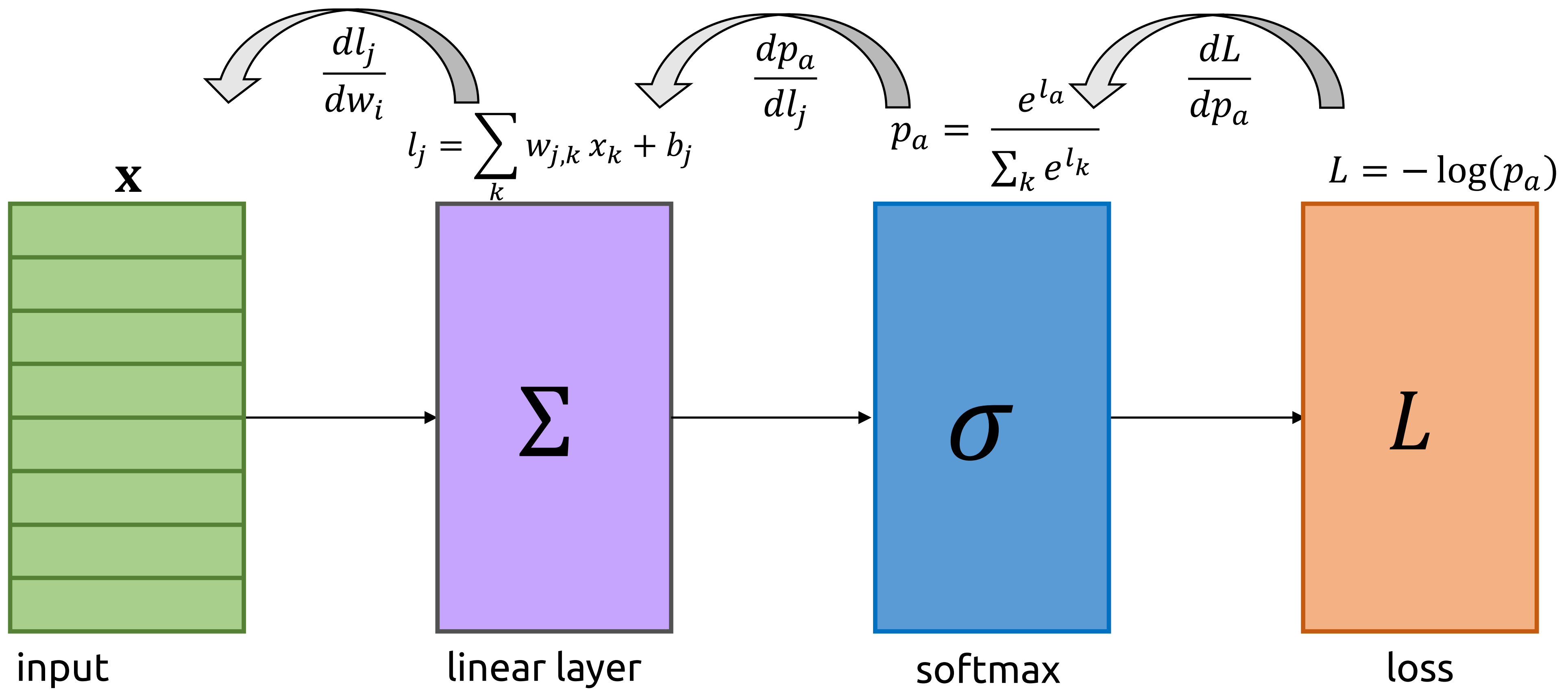
How can we calculate this gradient?

[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps]

Backpropagation is back!

How do we change this?

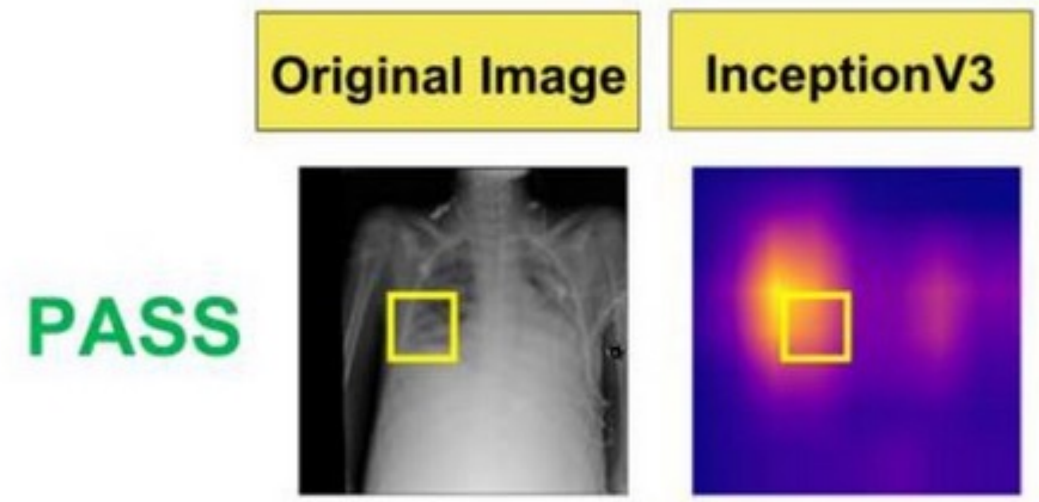
$$\Delta w_{j,i} = -\alpha \frac{\partial L}{\partial w_{j,i}} = -\alpha \cdot \frac{\partial L}{\partial p_a} \cdot \frac{\partial p_a}{\partial l_j} \cdot \frac{\partial l_j}{\partial w_{j,i}}$$



Saliency maps work well

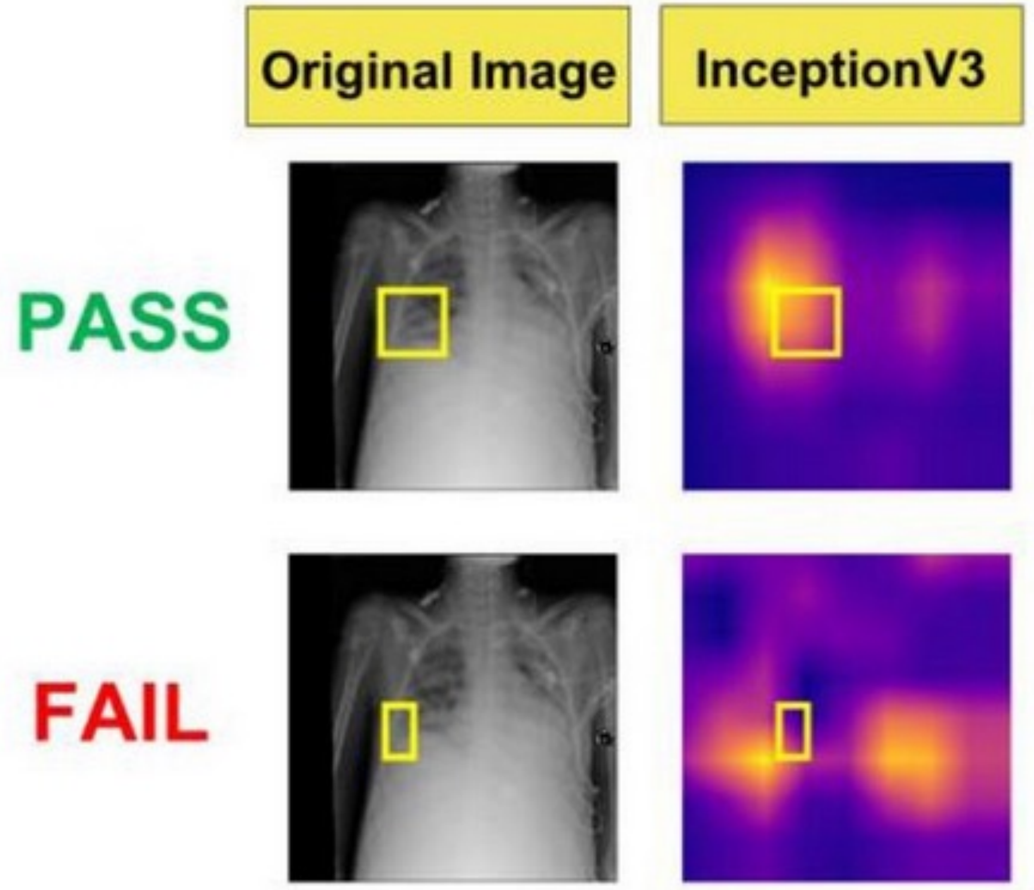
https://medium.datadriveninvestor.com/visualizing-neural-networks-using-saliency-maps-in-pytorch-289d8e244ab4

Saliency maps can also fail



[Assessing the (Un)Trustworthiness of Saliency Maps for Localizing Abnormalities in Medical Imaging]()

Saliency maps can also fail

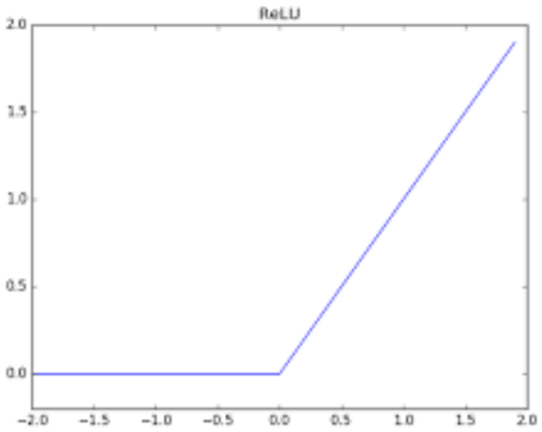


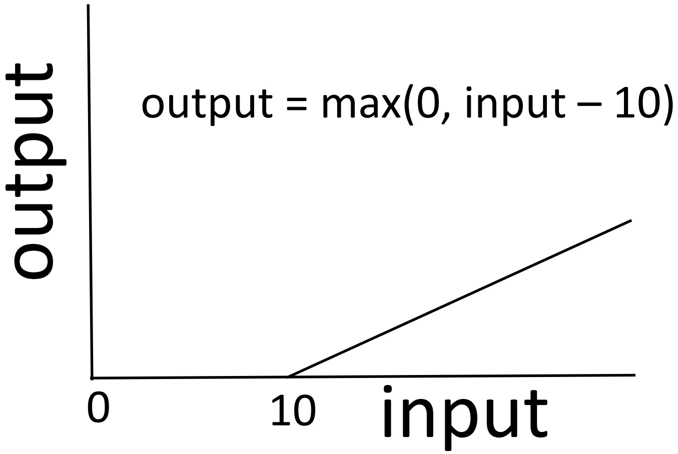
What could be going wrong?

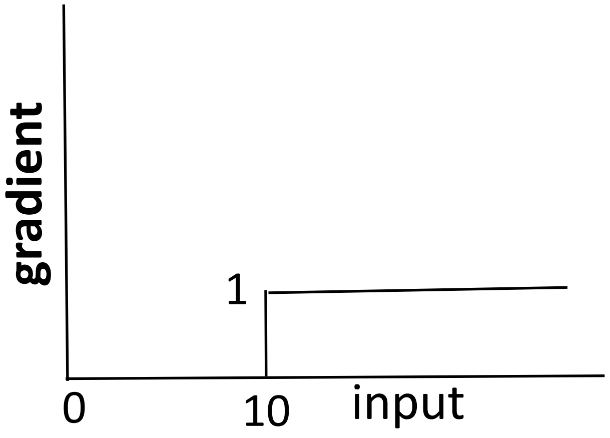
[Assessing the (Un)Trustworthiness of Saliency Maps for Localizing Abnormalities in Medical Imaging]

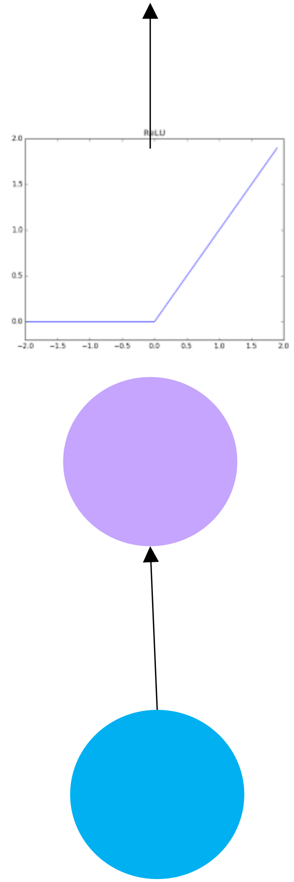
Backpropagation through activation functions

Thresholding problem





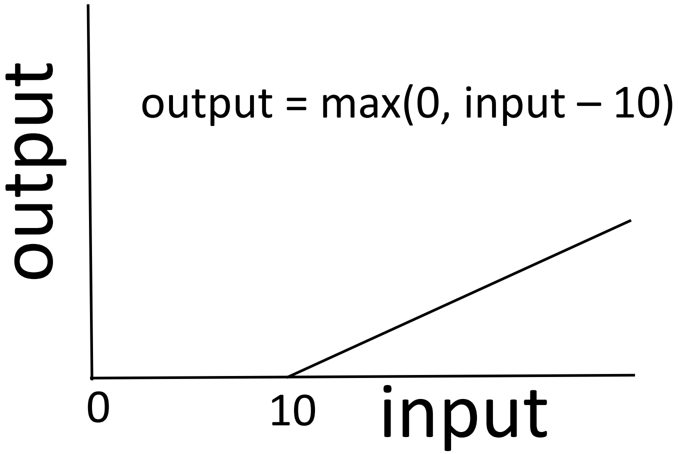


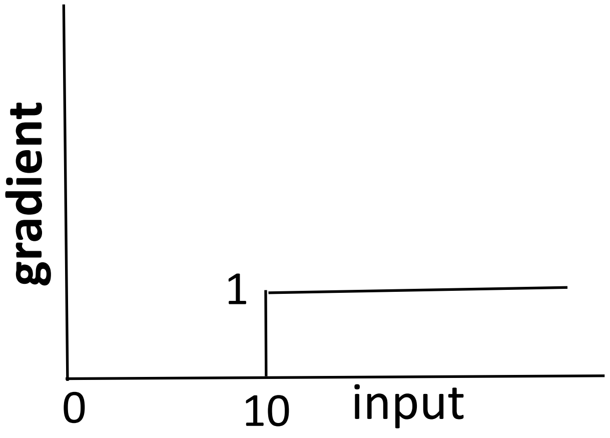


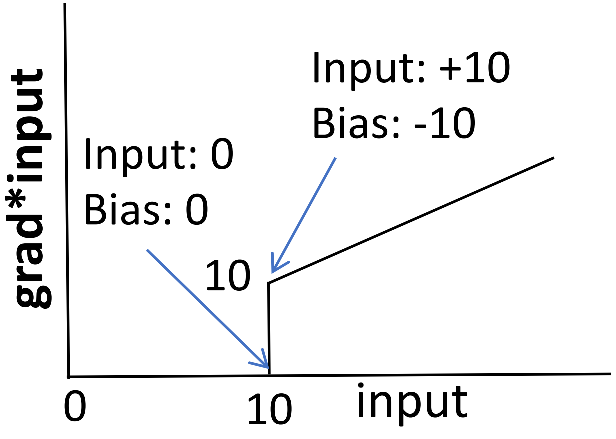
[Learning Important Features Through Propagating Activation Differences]

Any questions?

Gradient * Input

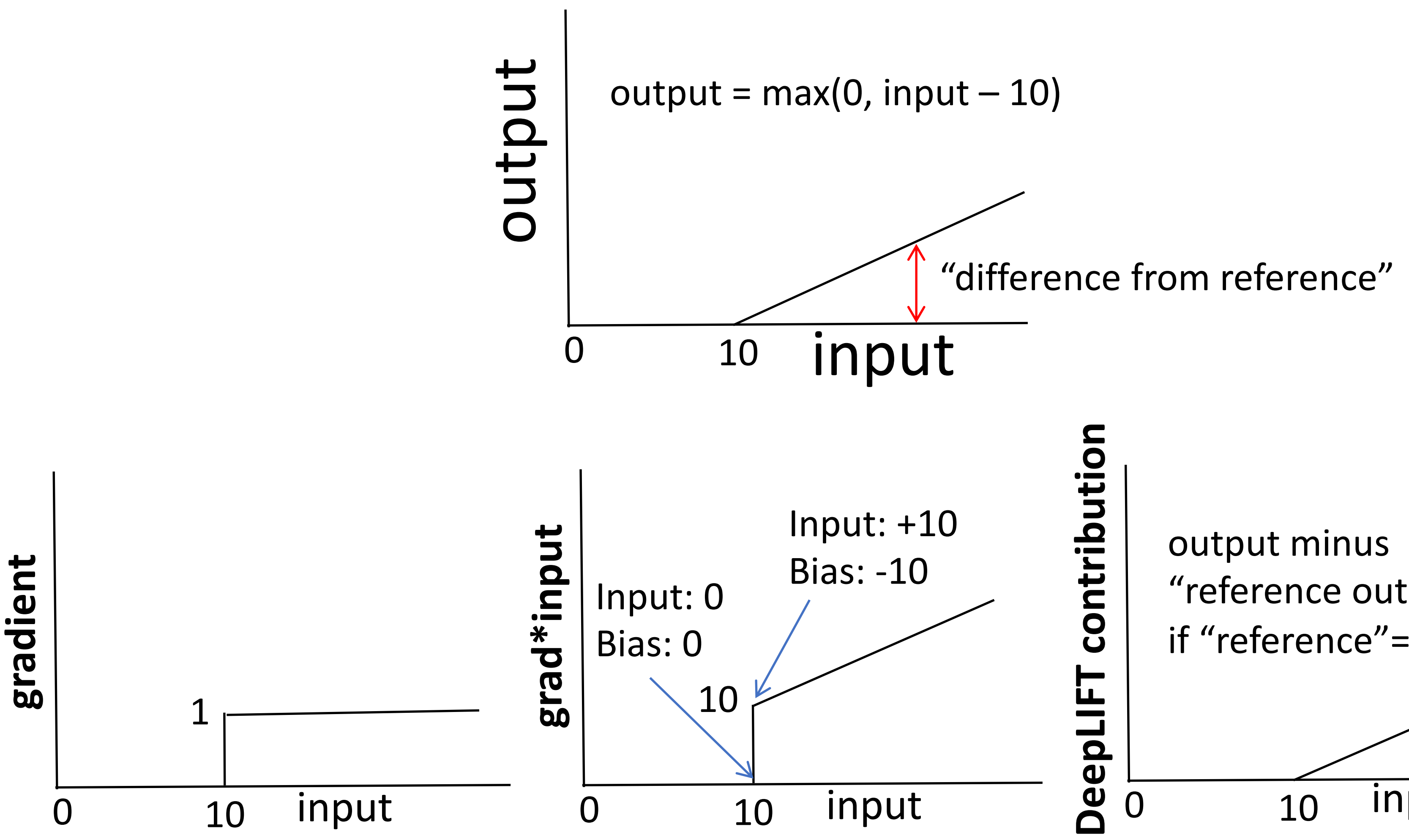






[Learning Important Features Through Propagating Activation Differences]()

Solves thresholding problem



Slide courtesy: https://drive.google.com/file/d/0B15F_QN41VQXbkVkcTVQYTVQNVE/view

Integrated gradients

Input image 

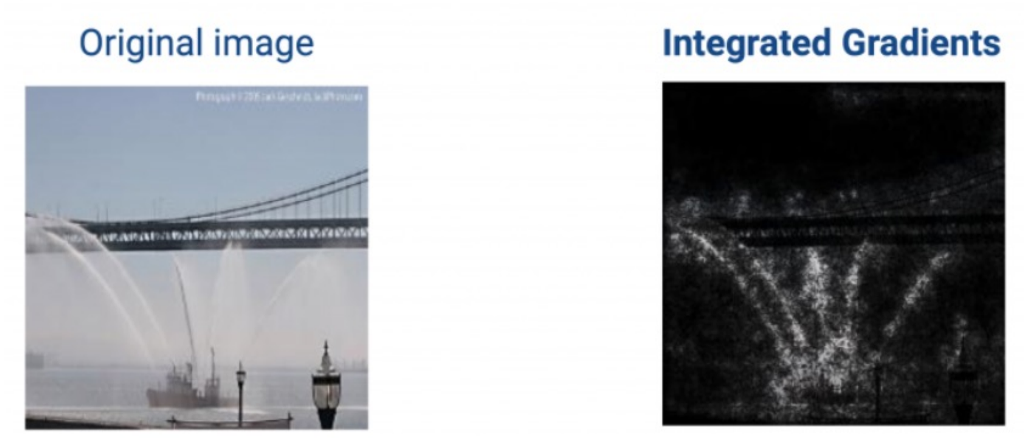
($\alpha = 1$)

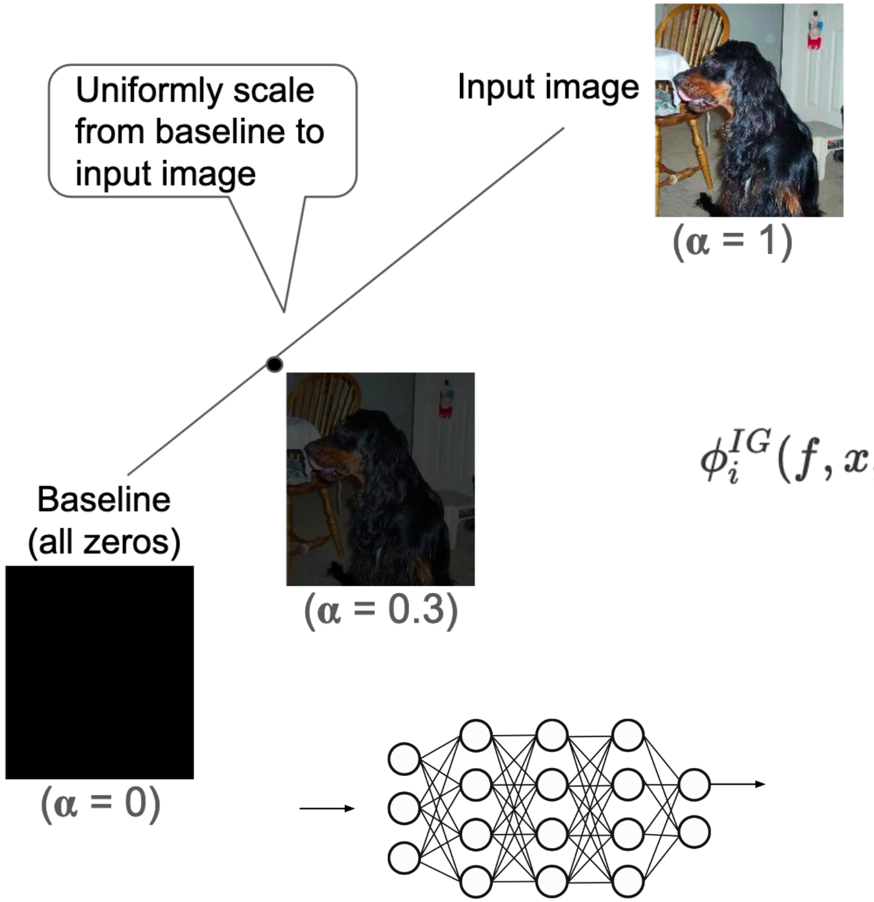
Baseline
(all zeros)

($\alpha = 0$)

Image courtesy: http://theory.stanford.edu/~ataly/Talks/sri_attribution_talk_jun_2017.pdf

Integrated gradients



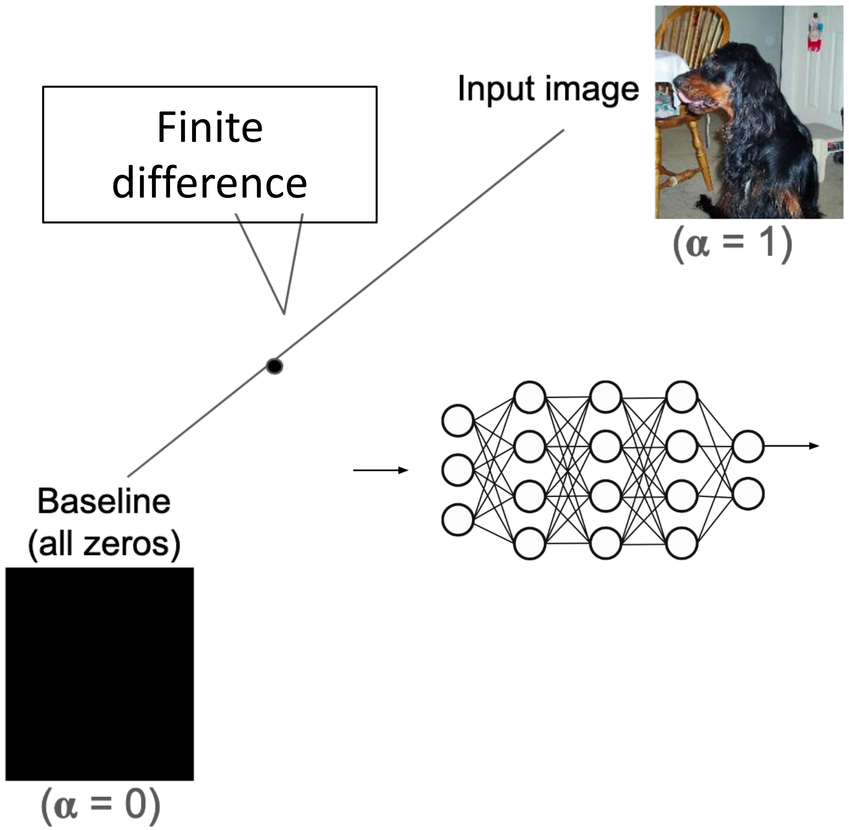


$$\phi_i^{IG}(f, x, x') = \overbrace{(x_i - x'_i)}^{\text{Difference from baseline}} \times \underbrace{\int_{\alpha=0}^{1}}_{\text{From baseline to input...}} \overbrace{\frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha}^{\text{...accumulate local gradients}}$$

Can we think of disadvantages of this method?

Image courtesy: http://theory.stanford.edu/~ataly/Talks/sri_attribution_talk_jun_2017.pdf

DeepLIFT

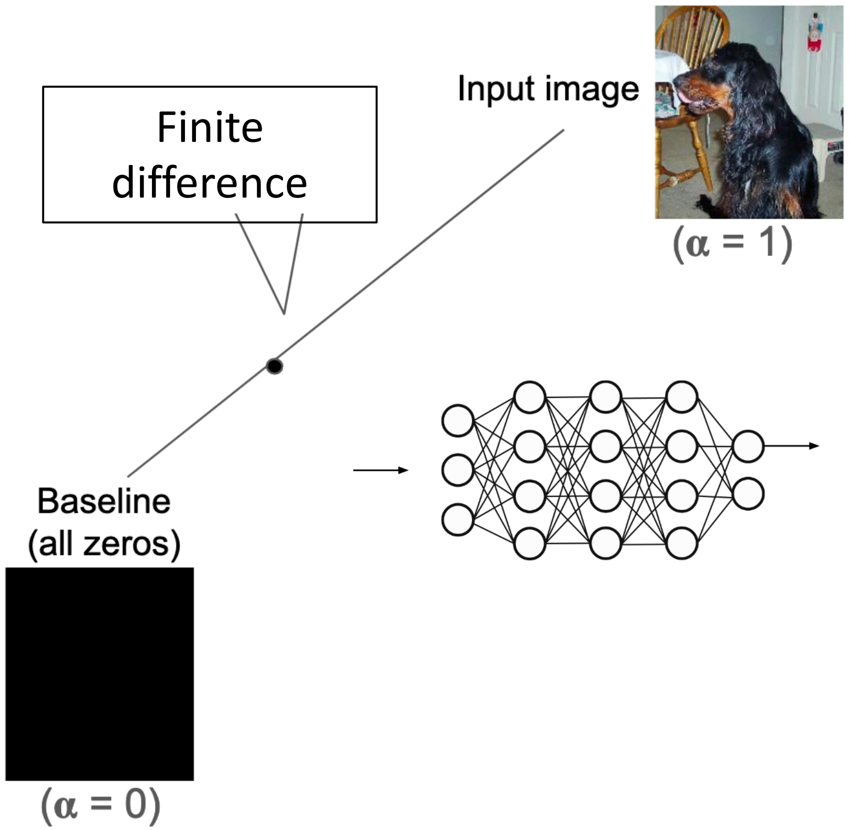


- Explain “difference from reference value” of output in terms of “difference from reference value” of inputs

Slide courtesy: https://drive.google.com/file/d/0B15F_QN41VQXbkVkcTVQYTVQNVE/view

[Learning Important Features Through Propagating Activation Differences]

DeepLIFT



- Explain “difference from reference value” of output in terms of “difference from reference value” of inputs
- Target neuron t with diff-from-ref Δt

Slide courtesy: https://drive.google.com/file/d/0B15F_QN41VQXbkVkcTVQYTVQNVE/view

[Learning Important Features Through Propagating Activation Differences]

https://github.com/slundberg/shap

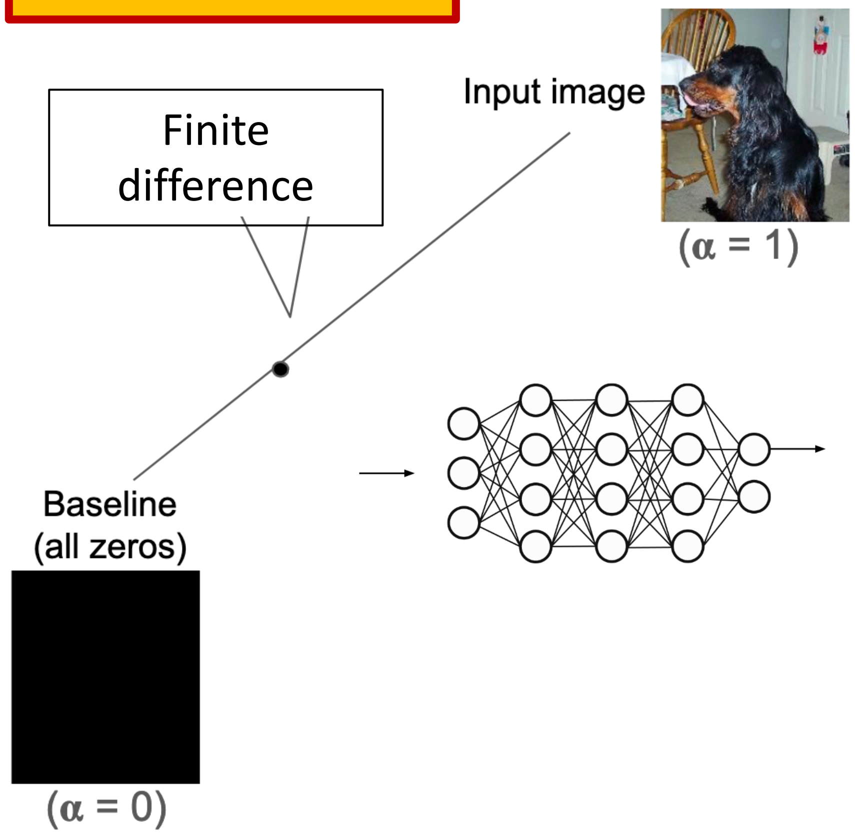
SHAP

shap.DeepExplainer

`class shap.DeepExplainer(model, data, session=None, learning_phase_flags=None)`

DeepLIFT

Can you think of any disadvantage of this method?



- Explain "difference from reference value" of output in terms of "difference from reference value" of inputs
- Target neuron t with diff-from-ref Δt

- "Blame" Δt on $\Delta x_1 \ldots \Delta x_n$
- Assign contributions $C_{\Delta x_i \Delta t}$ such that:

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta t} = \Delta t$$

Any questions?

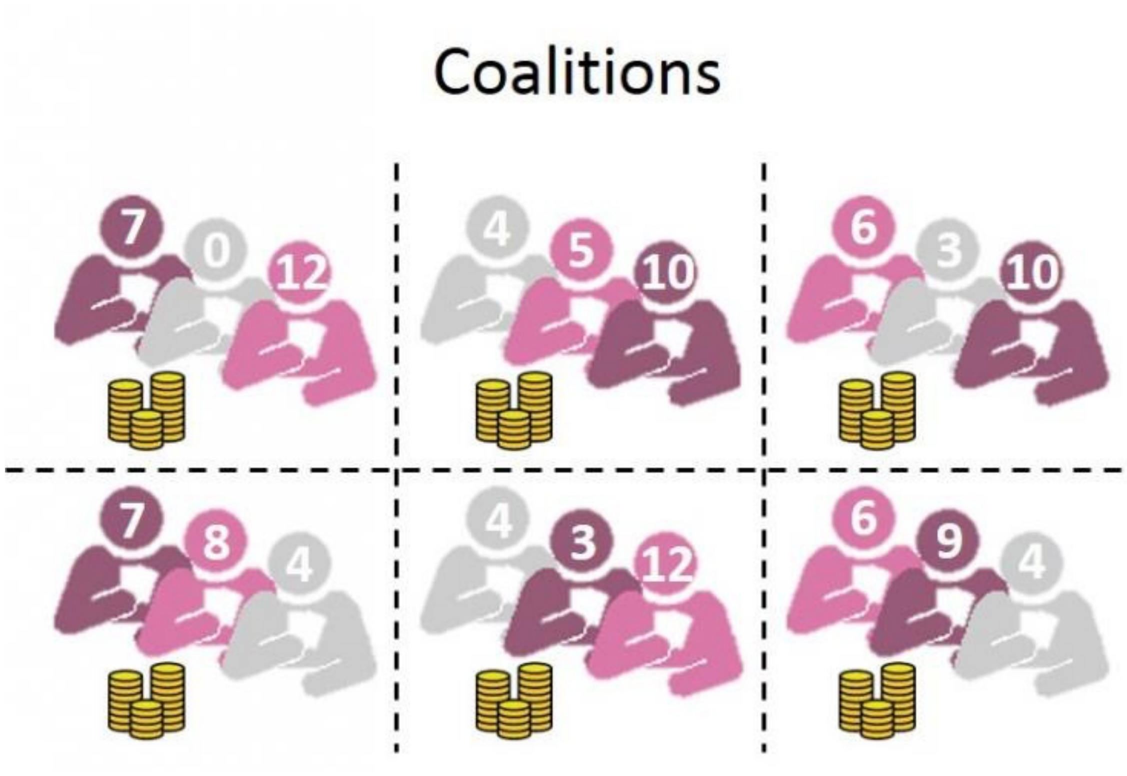
Slide courtesy: https://drive.google.com/file/d/0B15F_QN41VQXbkVkcTVQYTVQNVE/view

Learning Important Features Through Propagating Activation Differences

DeepLIFT (and Shapley values)

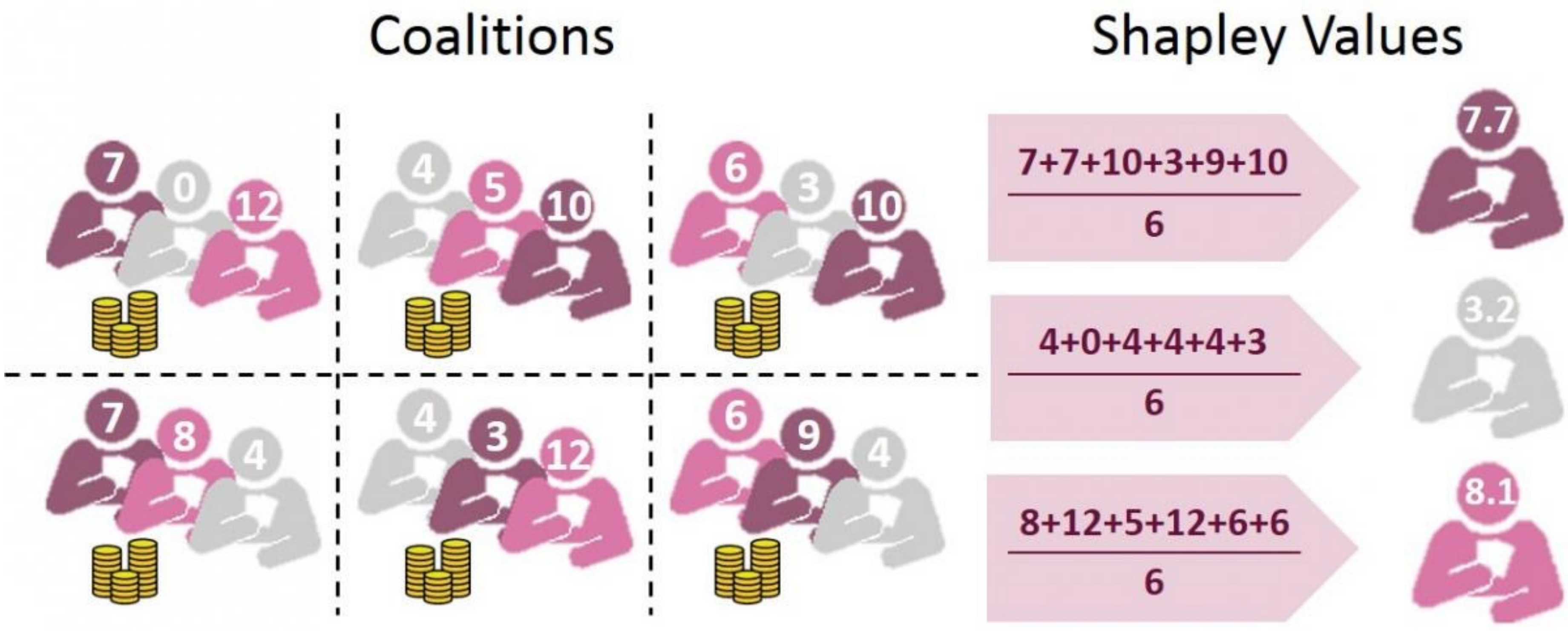
https://www.sia-partners.com/en/news-and-publications/from-our-experts/interpretable-machine-learning

DeepLIFT (and Shapley values)



https://www.sia-partners.com/en/news-and-publications/from-our-experts/interpretable-machine-learning

DeepLIFT (and Shapley values)



Shapley value not only considers the ability of each member, but also takes into account the cooperation among the members.

https://www.sia-partners.com/en/news-and-publications/from-our-experts/interpretable-machine-learning

Today’s goal – learn about interpretation in DL

(1) Model architecture based methods

(2) Gradient-based methods

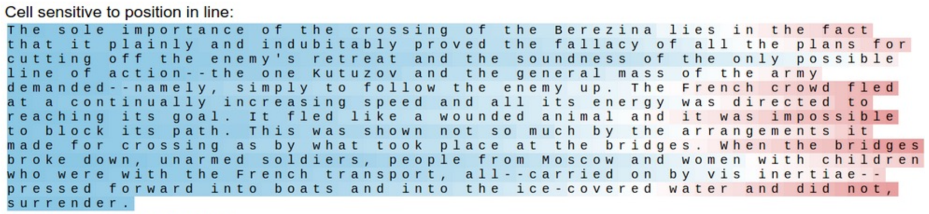
(3) Model agnostic methods

Next time!

Recap

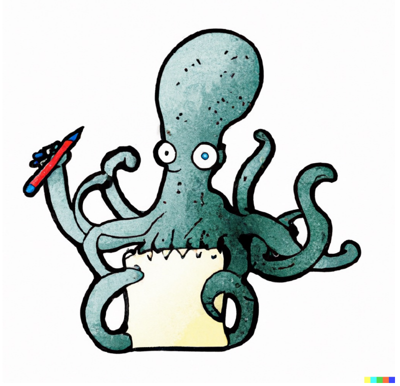
Model architecture based

- CNNs (Global Avg Pooling, Deconv)
- RNNs (Cell mapping, temporal output)



Gradient based and Shapley based

- Saliency map (Gradient * Input)
- Integrated gradients
- DeepLIFT



For in-depth reading refer to: https://christophm.github.io/interpretable-ml-book/