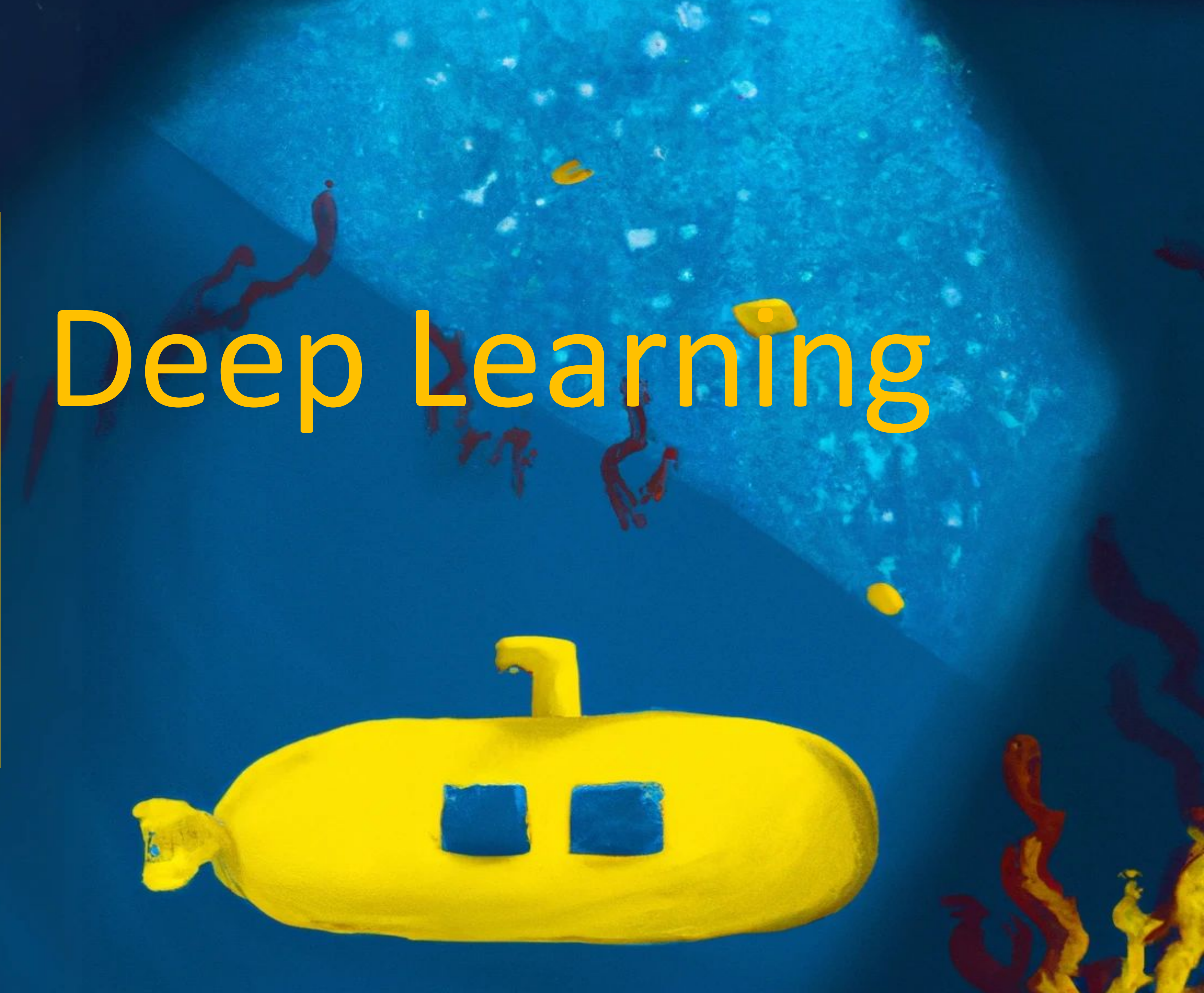
Deep Learning

CSCI 1470/2470
Spring 2023

Ritambhara Singh

March 22, 2023
Wednesday



DALL-E 2 prompt “a painting of deep underwater with a yellow submarine in the bottom right corner”

Machine Learning and Interpretability

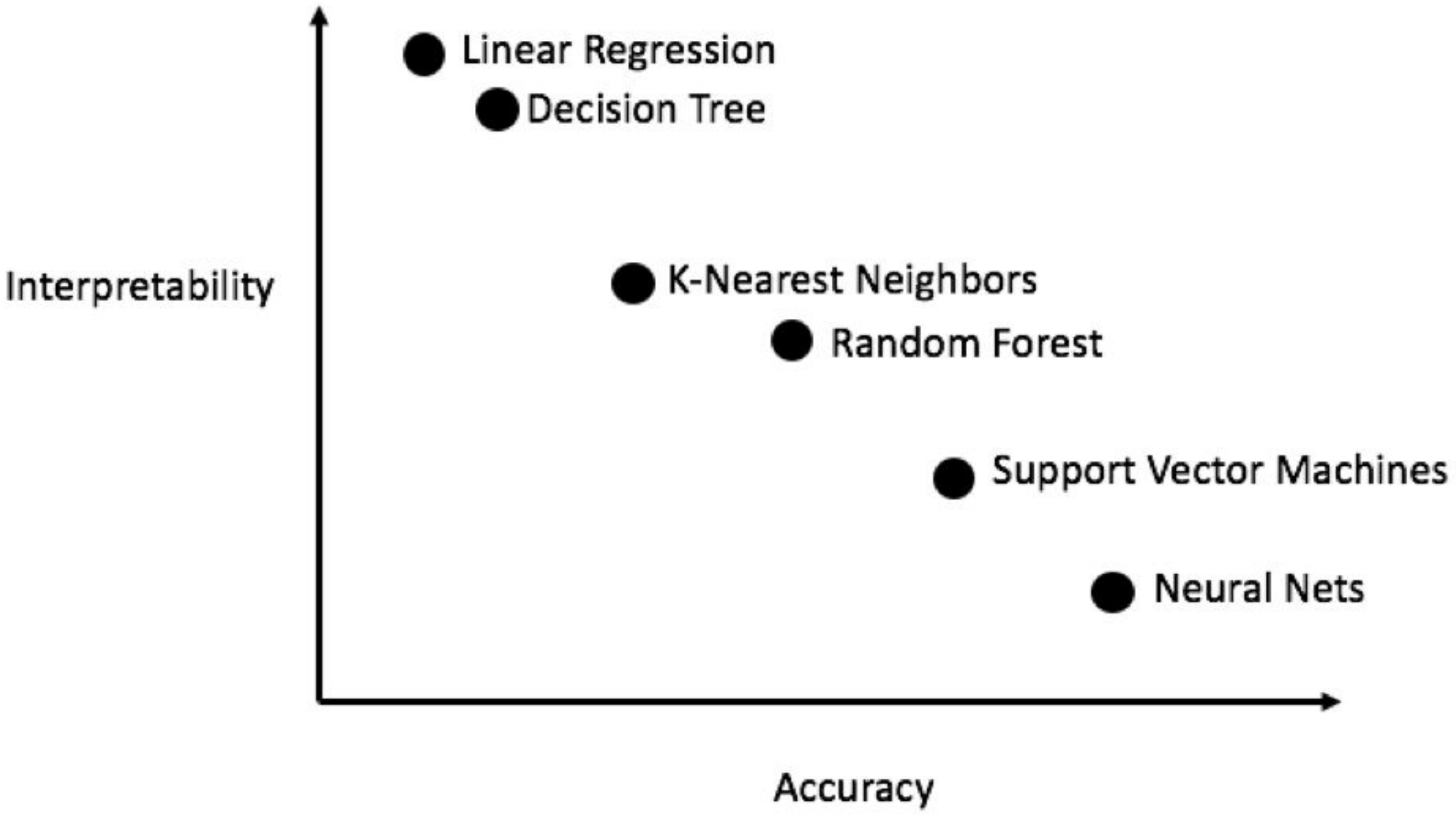
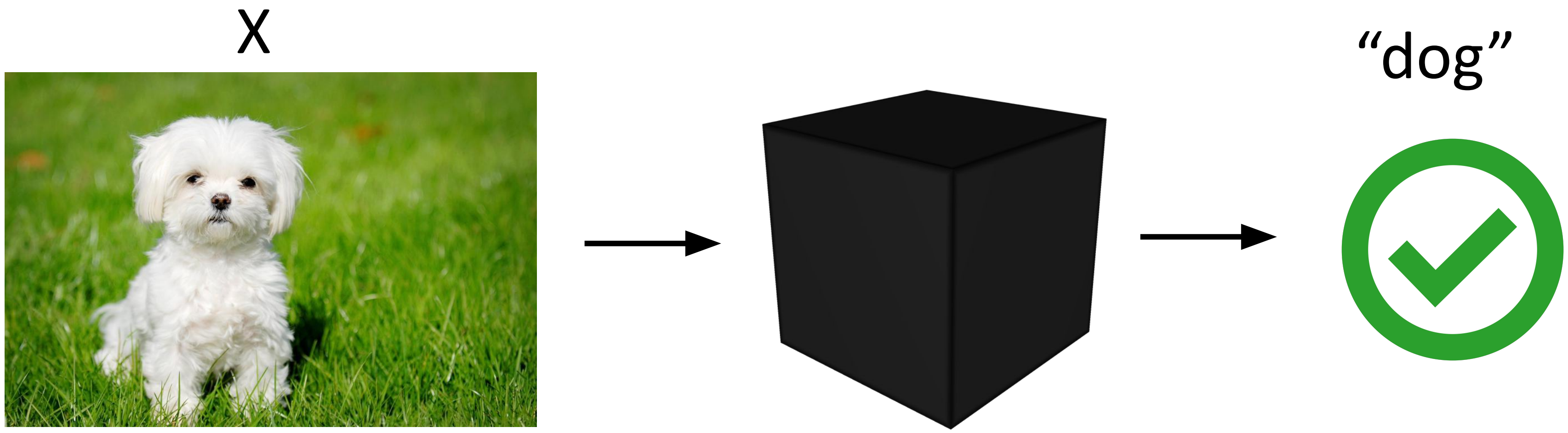


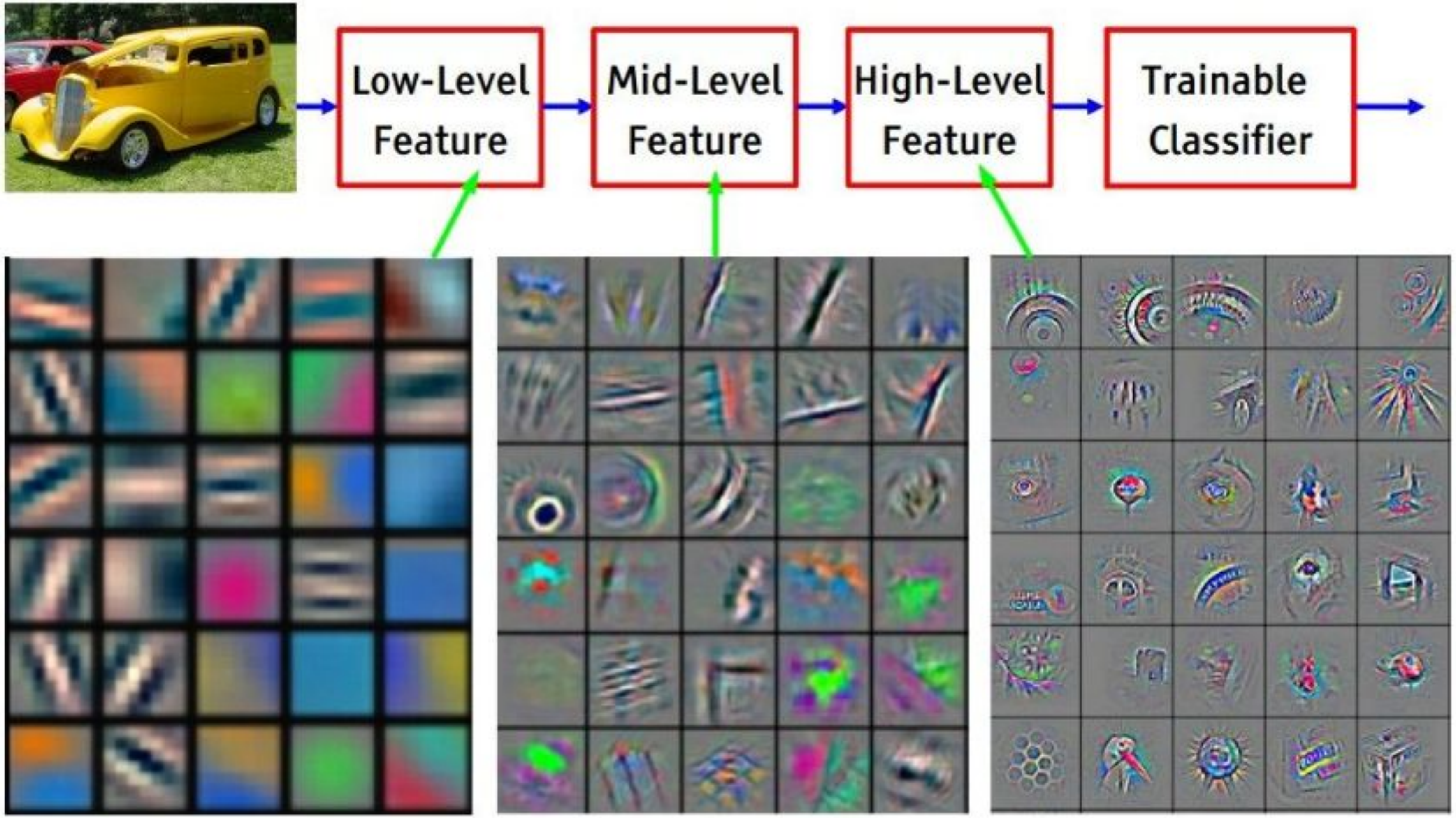
Image courtesy: https://medium.com/ansaro-blog/interpreting-machine-learning-models-1234d735d6c9

Deep Nets are “Black Box” Models

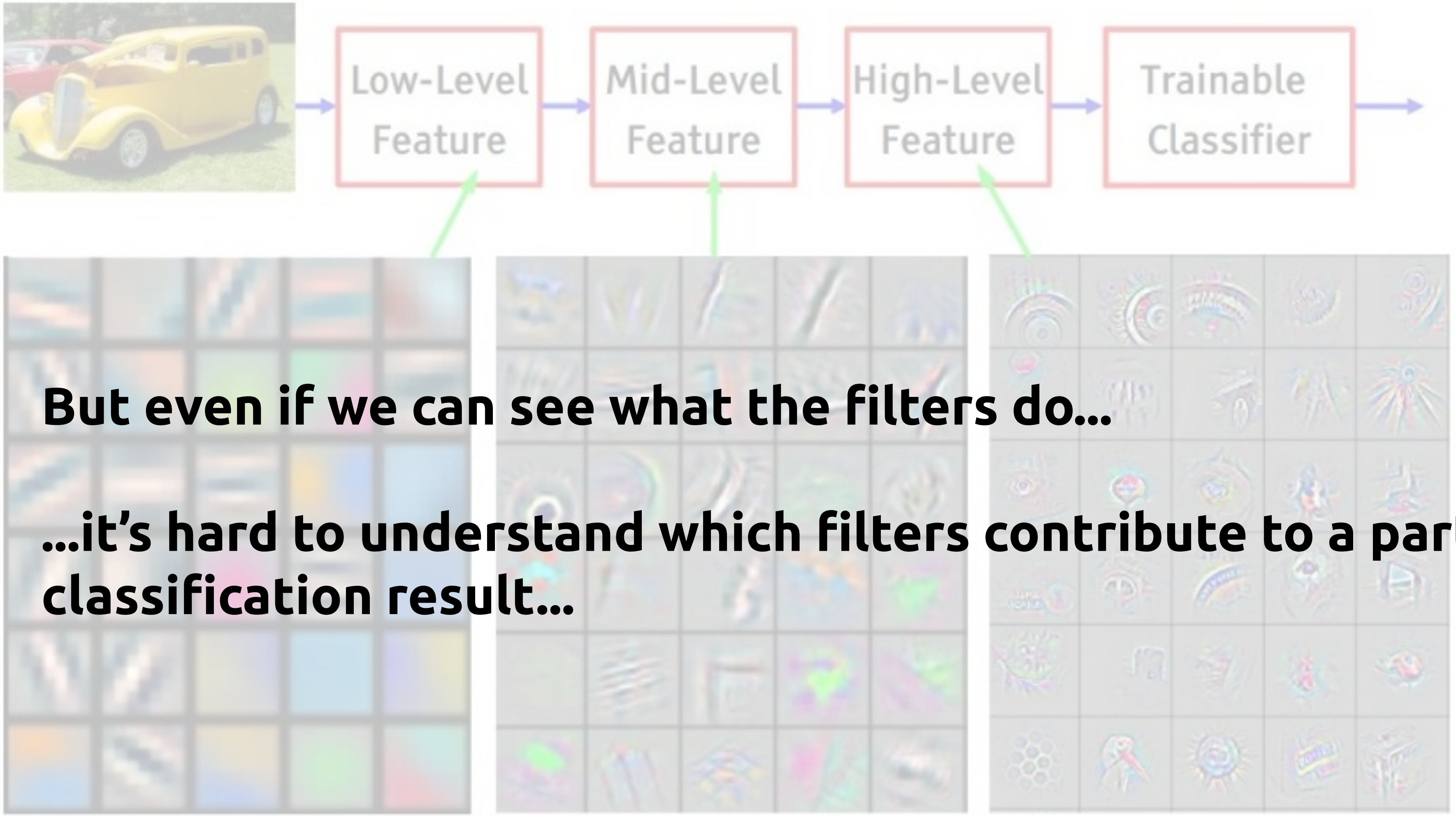
- What do the hidden layers of networks actually learn?
  - **Image recognition:** What do the many thousands of filters actually represent??
  - **Natural language processing**: What does the RNN hidden state actually store??



Example: What do CNN filters look like?



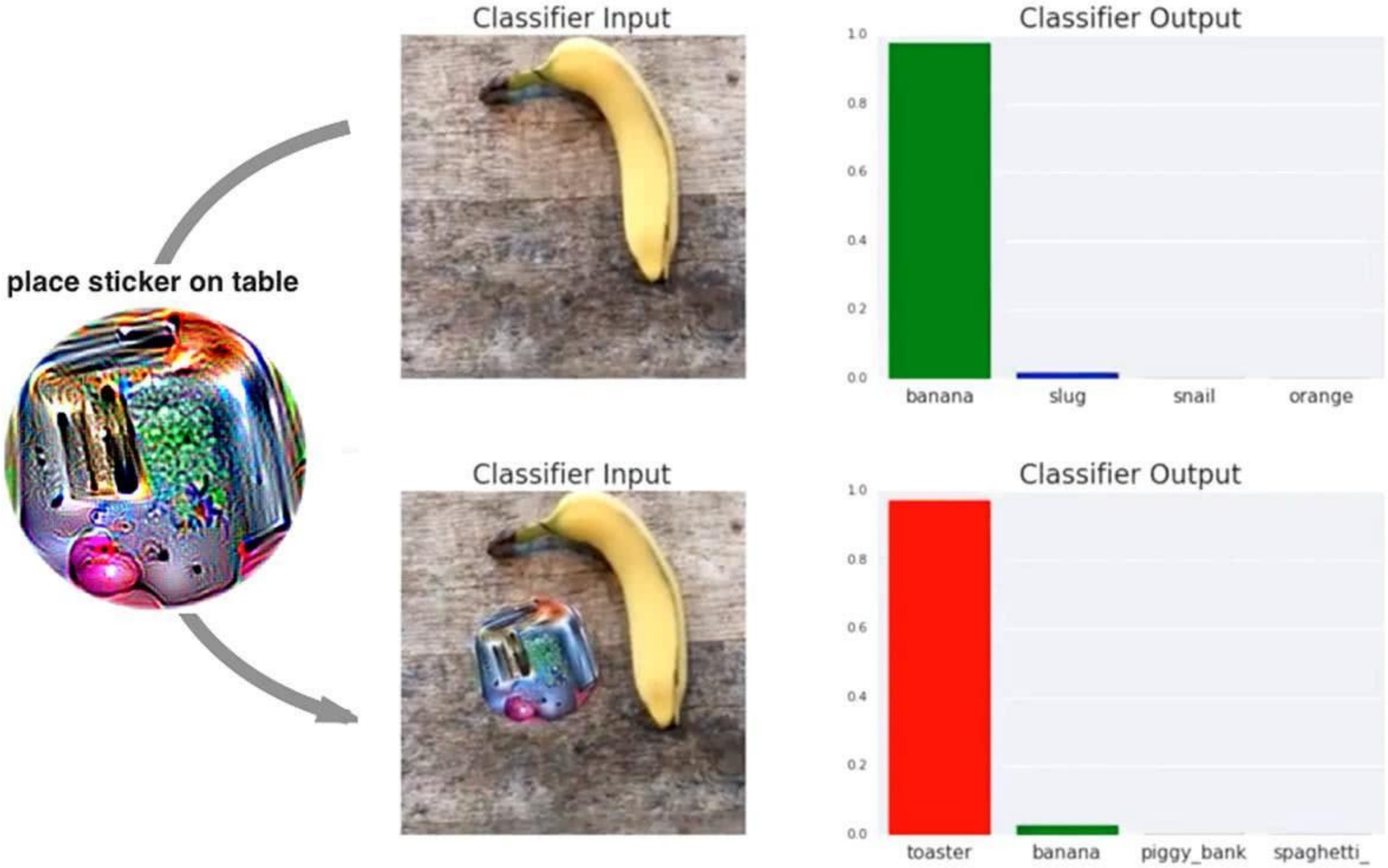
Example: What do CNN filters look like?



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:

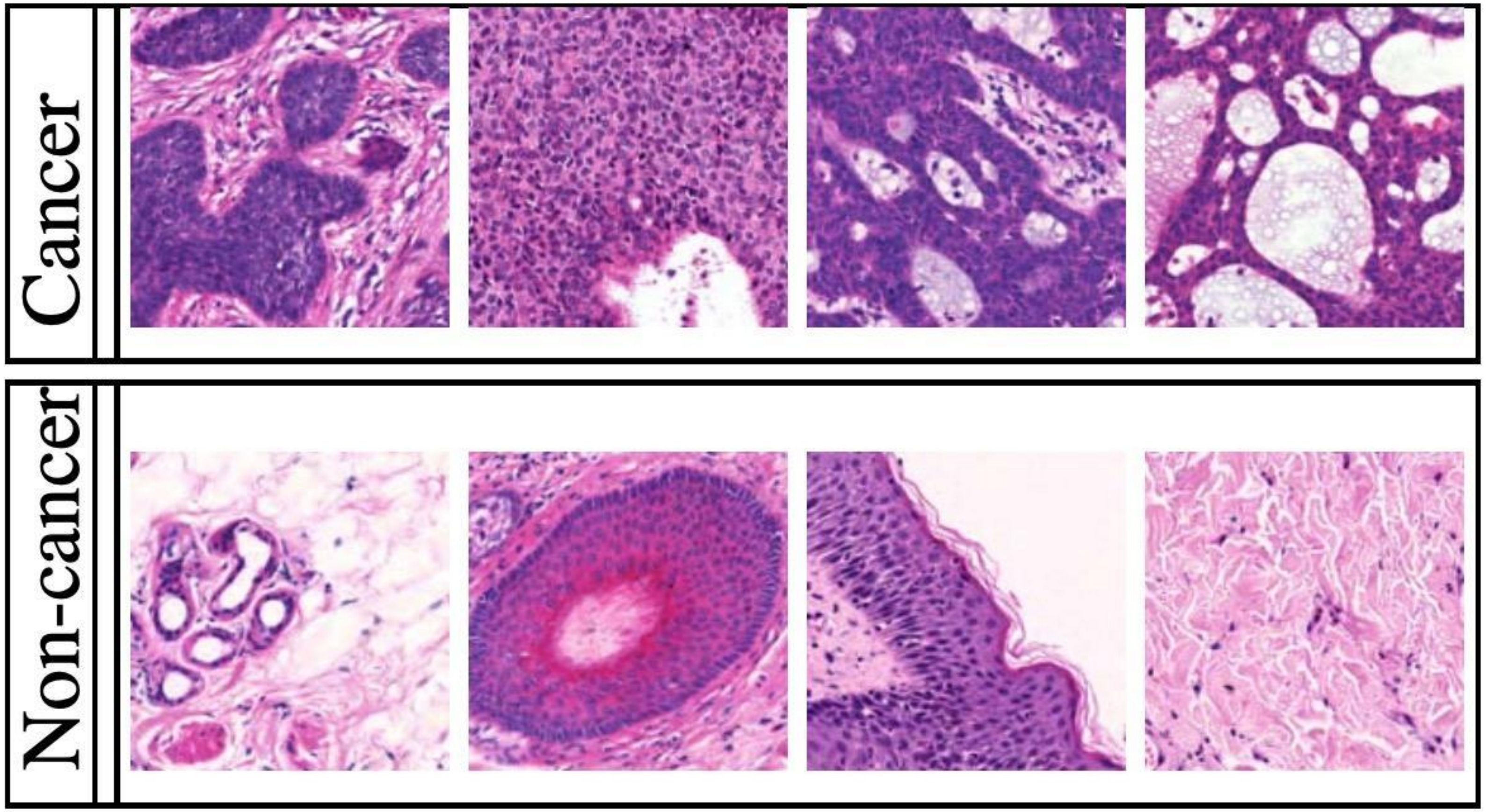


Deep Learning has an ***interpretability*** problem

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)?

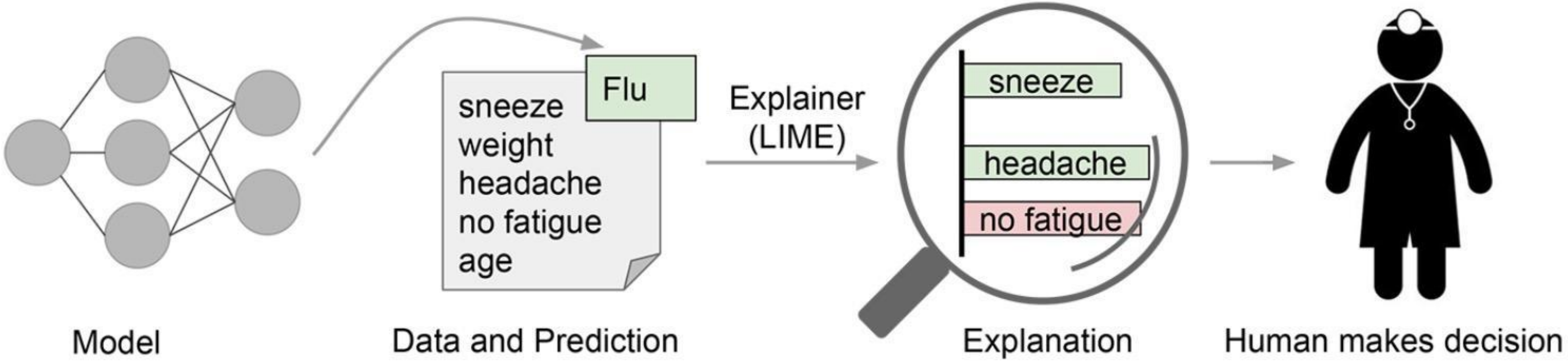


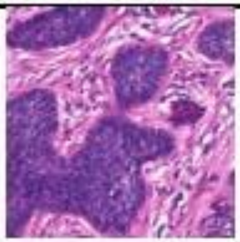
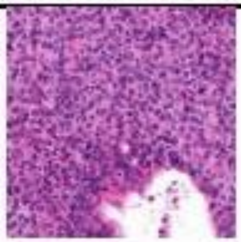
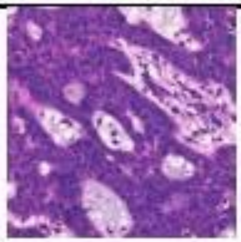
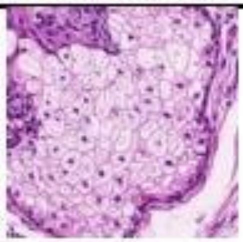
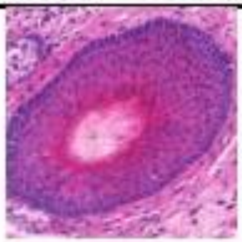
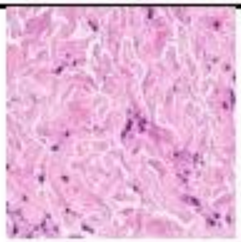
Figure 1. Explaining individual predictions to a human decision-maker. Source: Marco Tulio Ribeiro.

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 emerging subfield of study: ***Interpretable DL/AI***

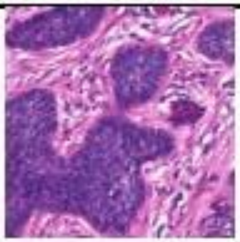
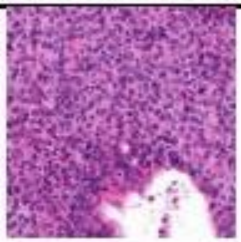
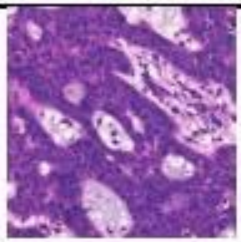
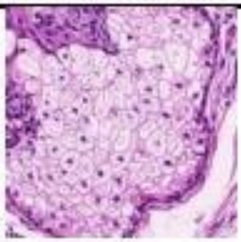
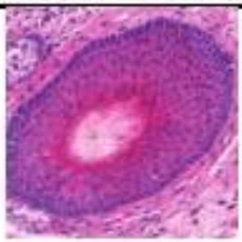
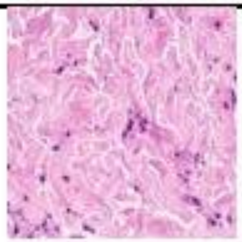
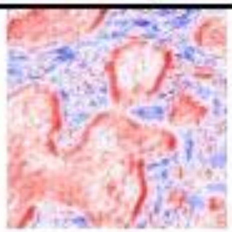
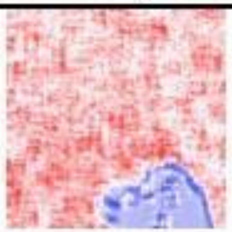
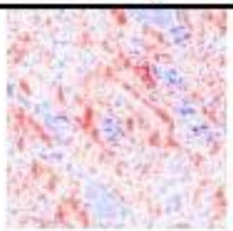
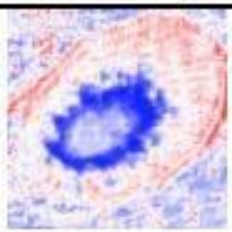
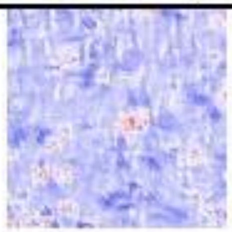
Making Cancer Predicting Interpretable

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)

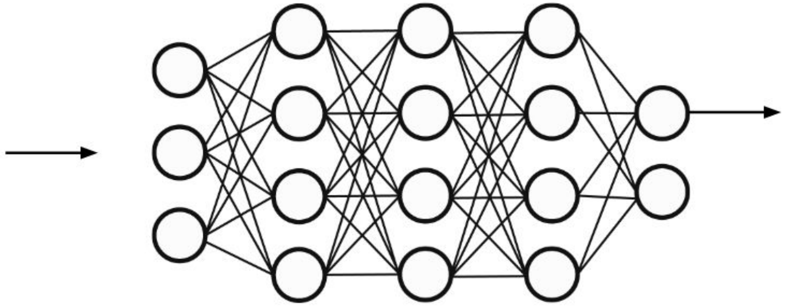
Making Cancer Predicting Interpretable

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

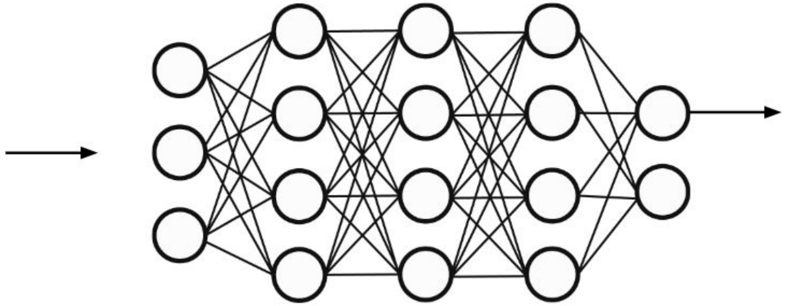
Task formulation

X  Y

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

Task formulation

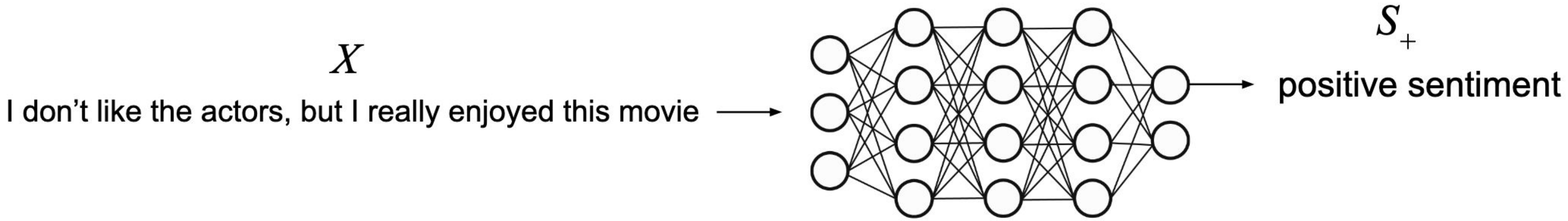
X



“dog”

Which pixels are most important for classification?

Task formulation



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)

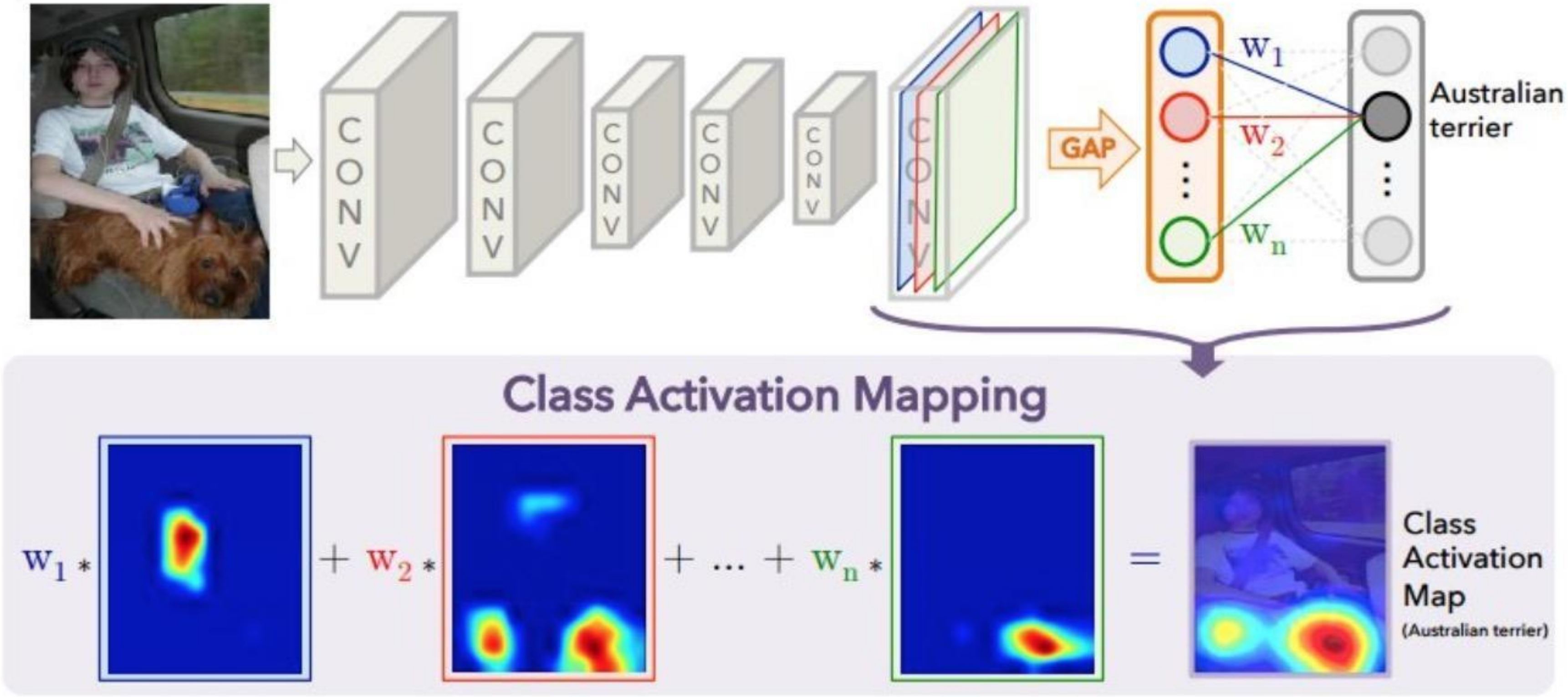
(2) Gradient-based methods

(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

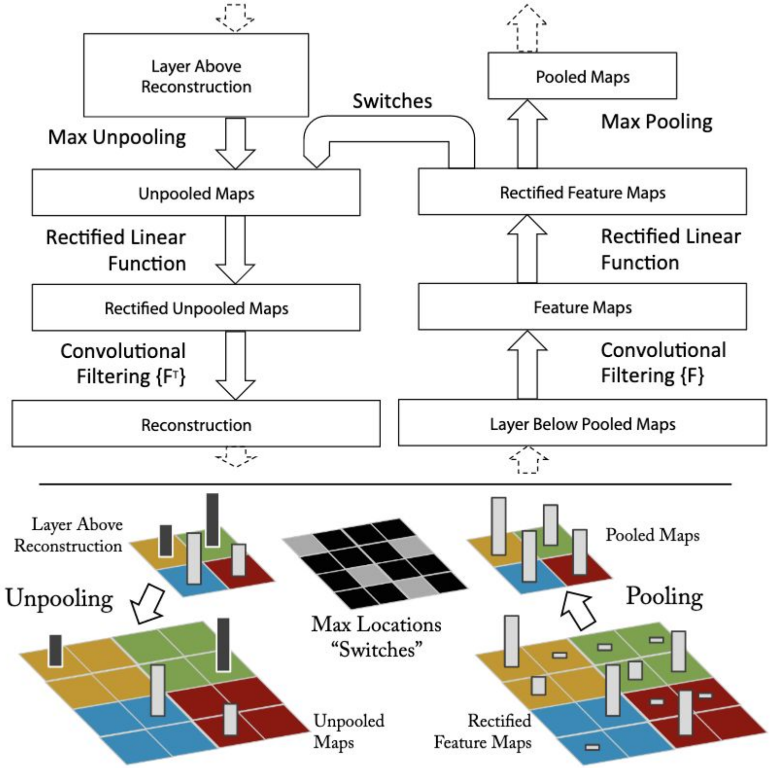


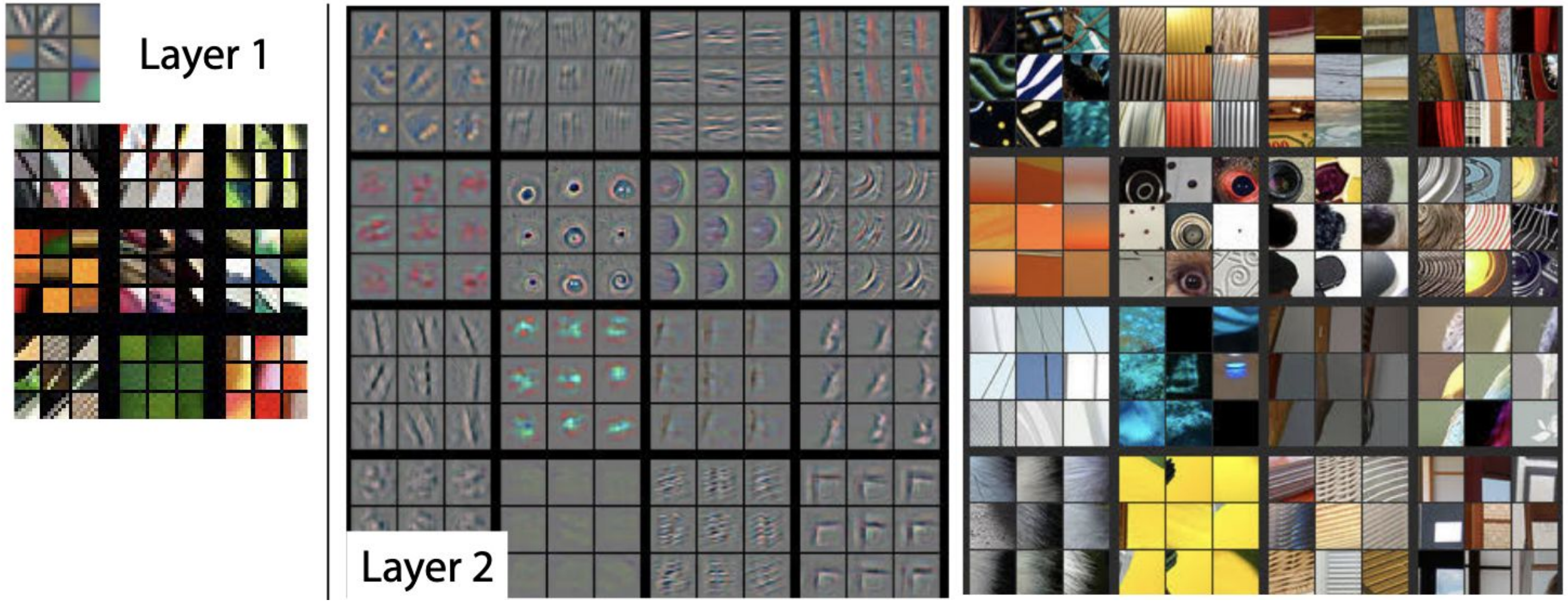
Deconvolution

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.



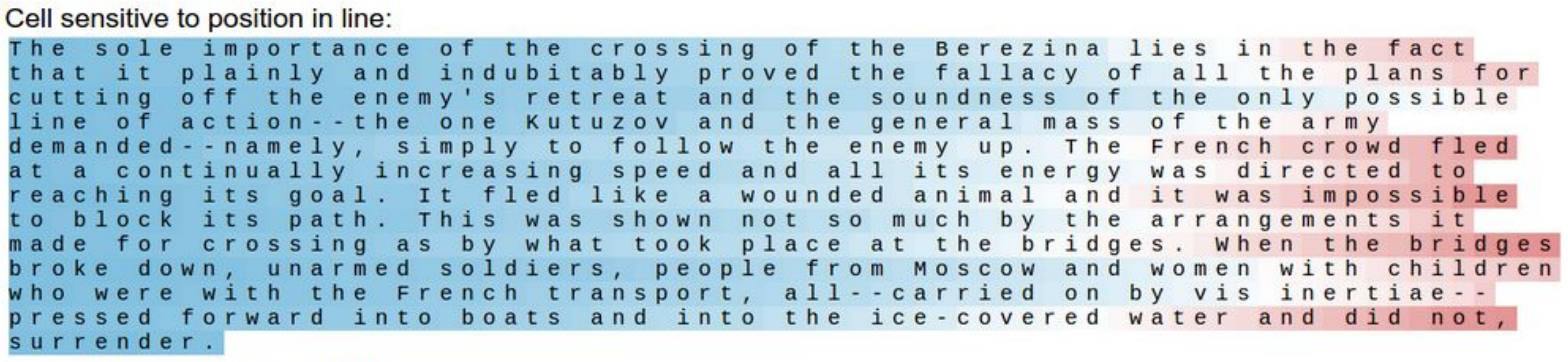


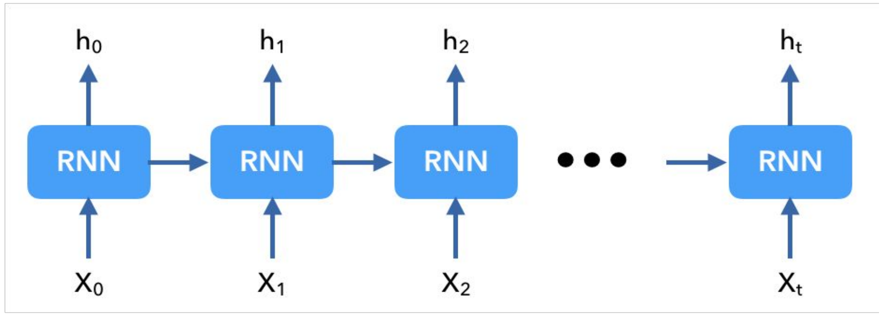
Visualizing and Understanding Convolutional Networks

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



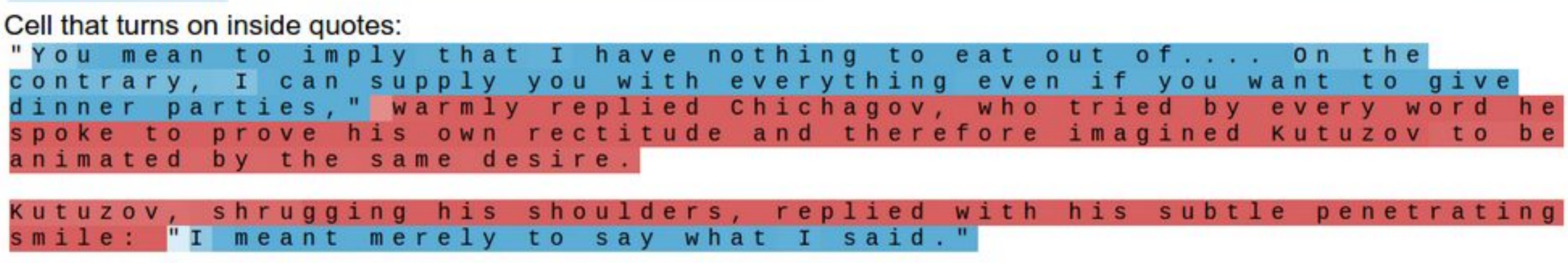


http://karpathy.github.io/2015/05/21/rnn-eff

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

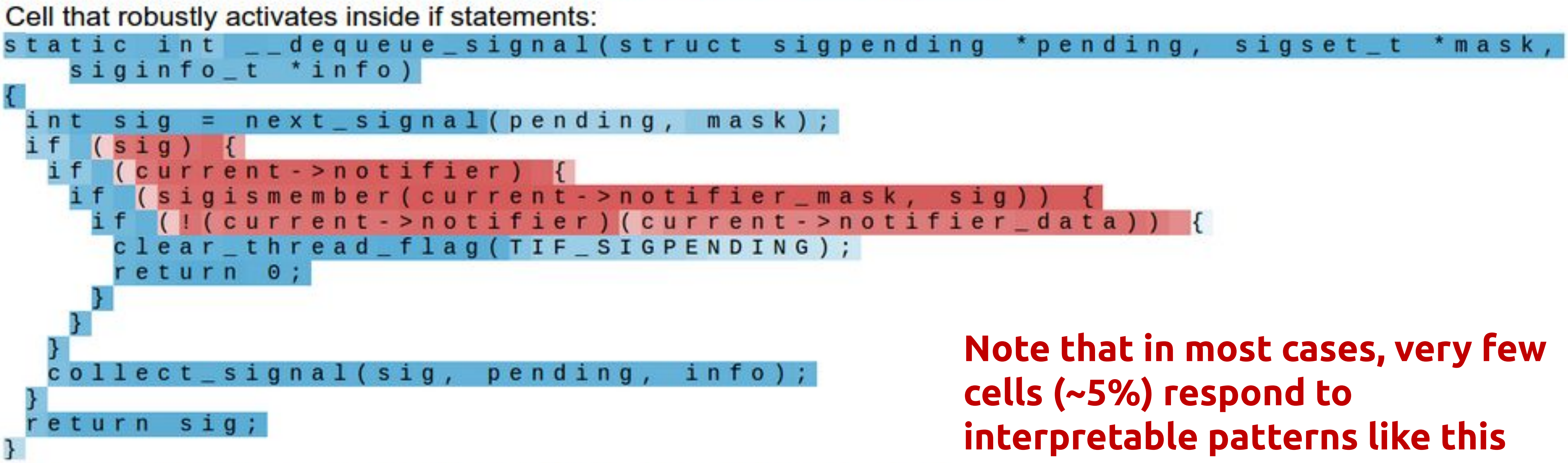


http://karpathy.github.io/2015/05/21/rnn-eff

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



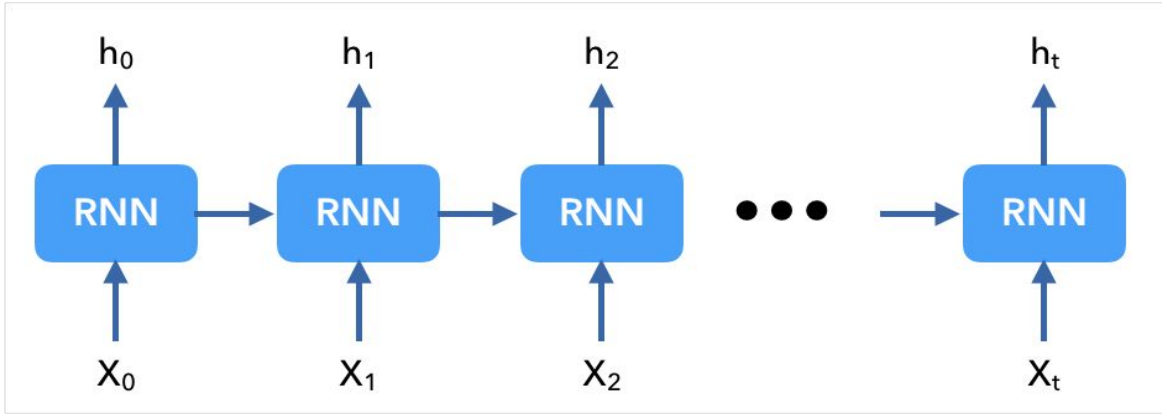
Note that in most cases, very few cells (~5%) respond to interpretable patterns like this

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

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

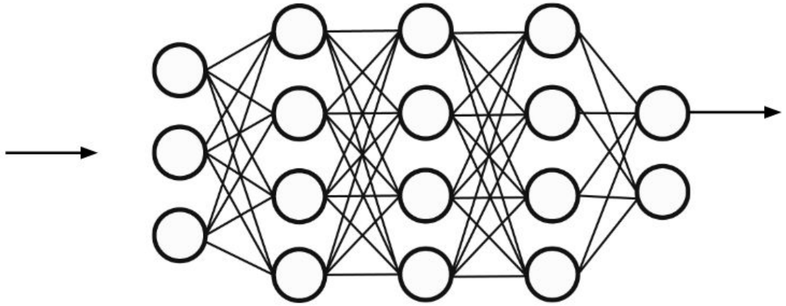
(1) Model architecture based methods

(2) Gradient-based methods

(3) Model agnostic methods

Saliency maps

X

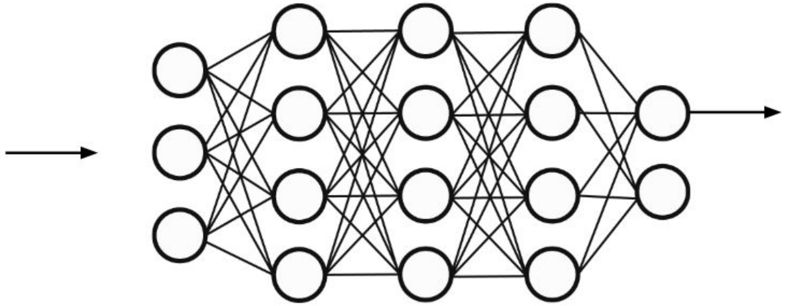


“dog”

Which pixels are most important for classification?

Saliency maps

X



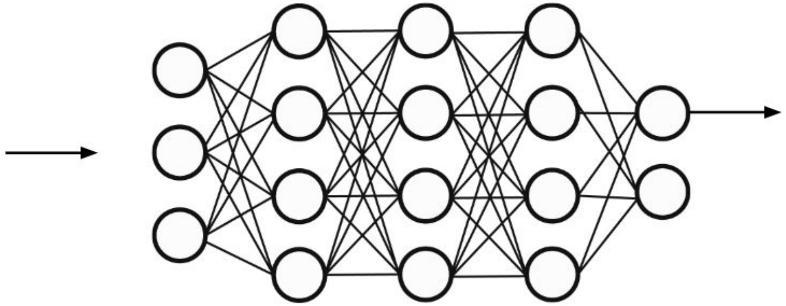
$S_+(X)$

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

<u>**Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps**</u>

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

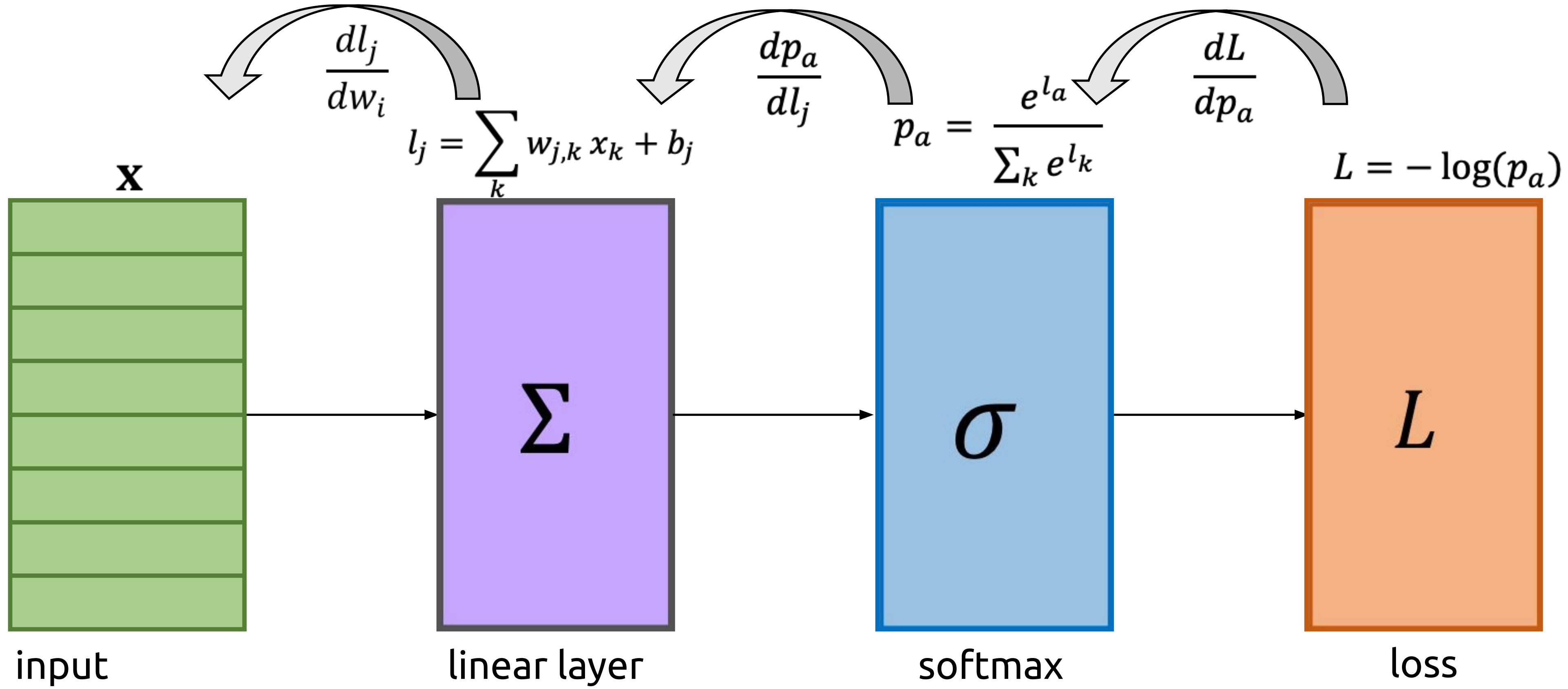
Backpropagation is back!

- The process of calculating gradients of functions via chain rule in a neural network
- Is a part of and **NOT the whole learning algorithm**
- **Can be calculated with respect to any variable of choice**
- For learning in neural networks we calculate gradients with respect to the weights

Backpropagation is back!

How do we change this?

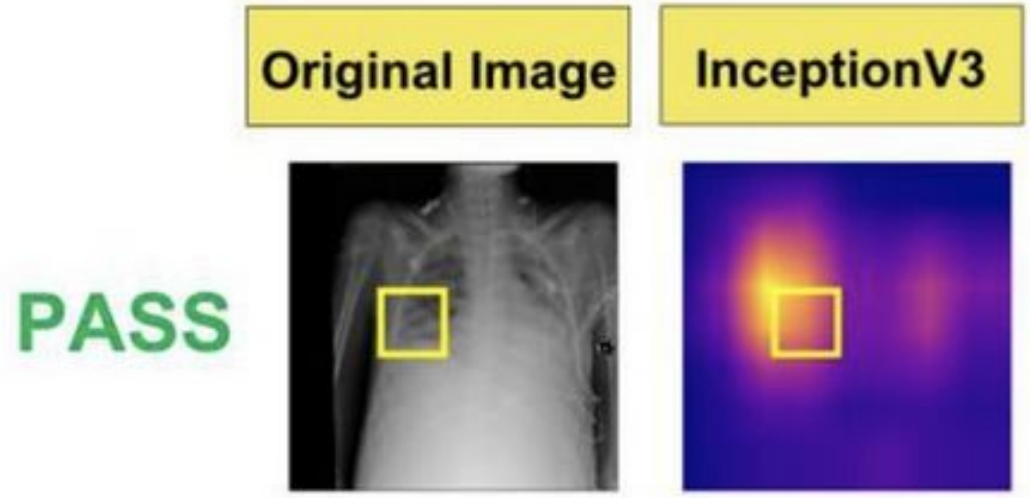
$$\Delta w_{j,i} = -\alpha \frac{\partial L}{\partial w_{j,i}} = -\alpha \cdot \frac{\partial L}{\partial p_a} \cdot \frac{\partial p_a}{\partial l_j} \cdot \frac{\partial l_j}{\partial w_{j,i}}$$



Saliency maps work well

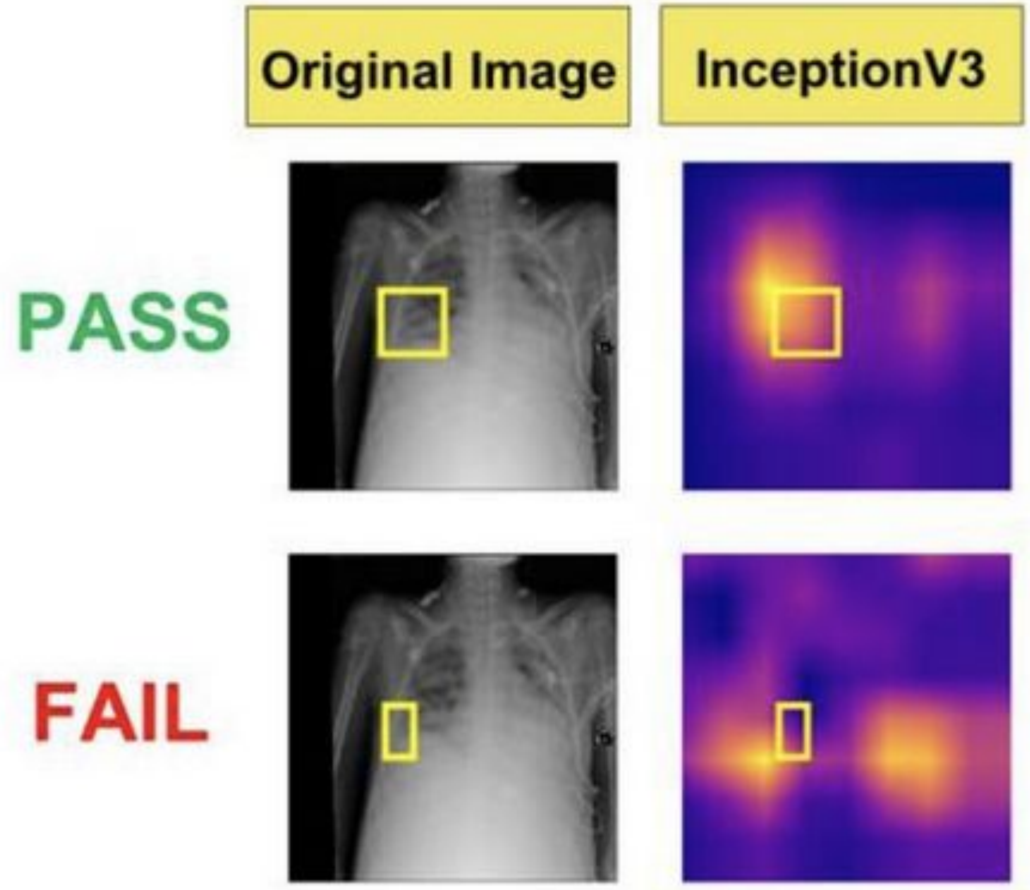
https://medium.datadriveninvestor.com/visualizing-neural-networks-using-saliency-maps-in-pytorch-289d8e244ab4

Saliency maps can also fail



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

Saliency maps can also fail

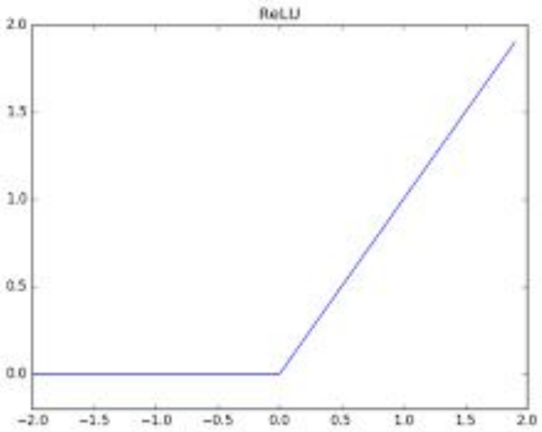


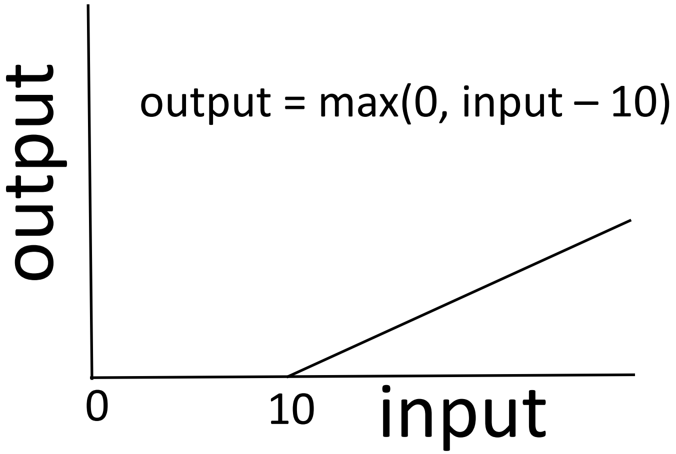
What could be going wrong?

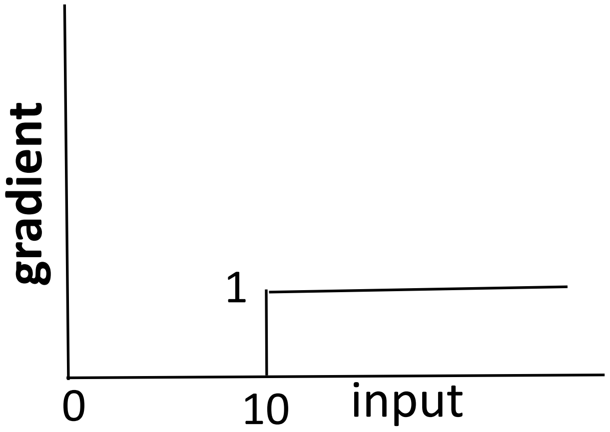
Assessing the (Un)Trustworthiness of Saliency Maps for Localizing Abnormalities in Medical Imaging

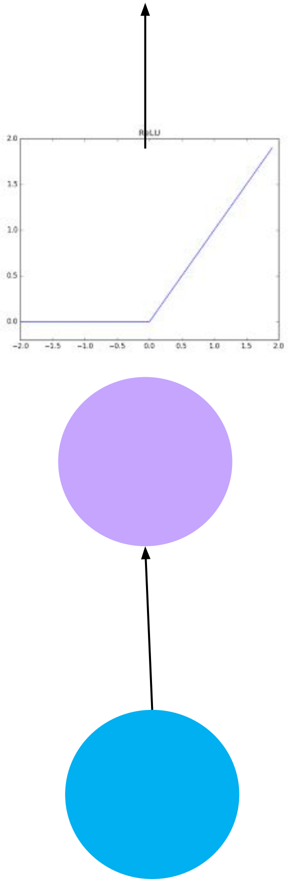
Backpropagation through activation

Thresholding problem





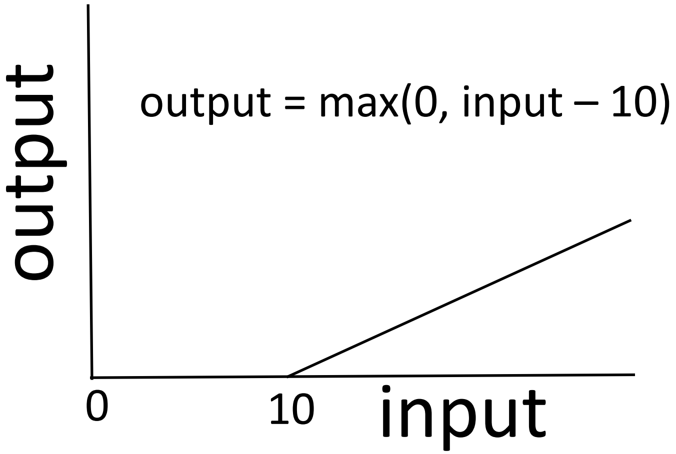


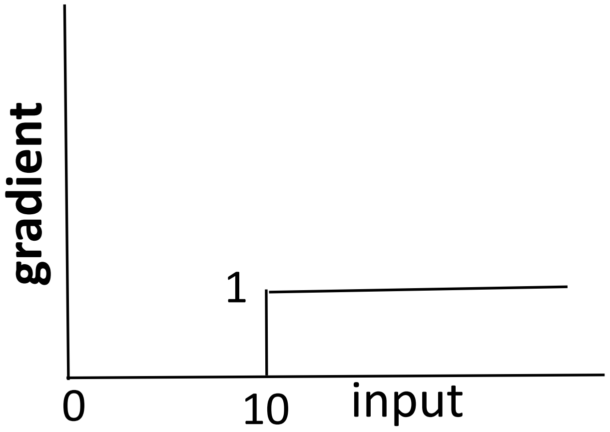


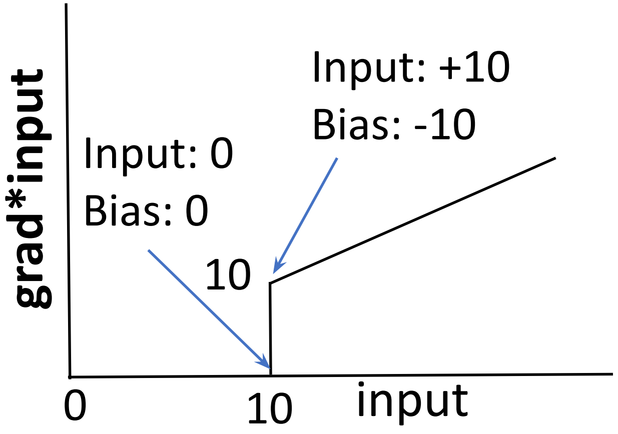
Learning Important Features Through Propagating Activation Differences

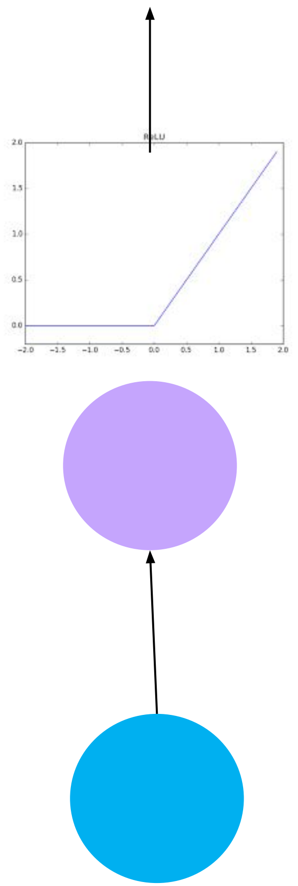
Gradient * Input

Any questions?









Learning Important Features Through Propagating Activation Differences

Integrated gradients

Input image

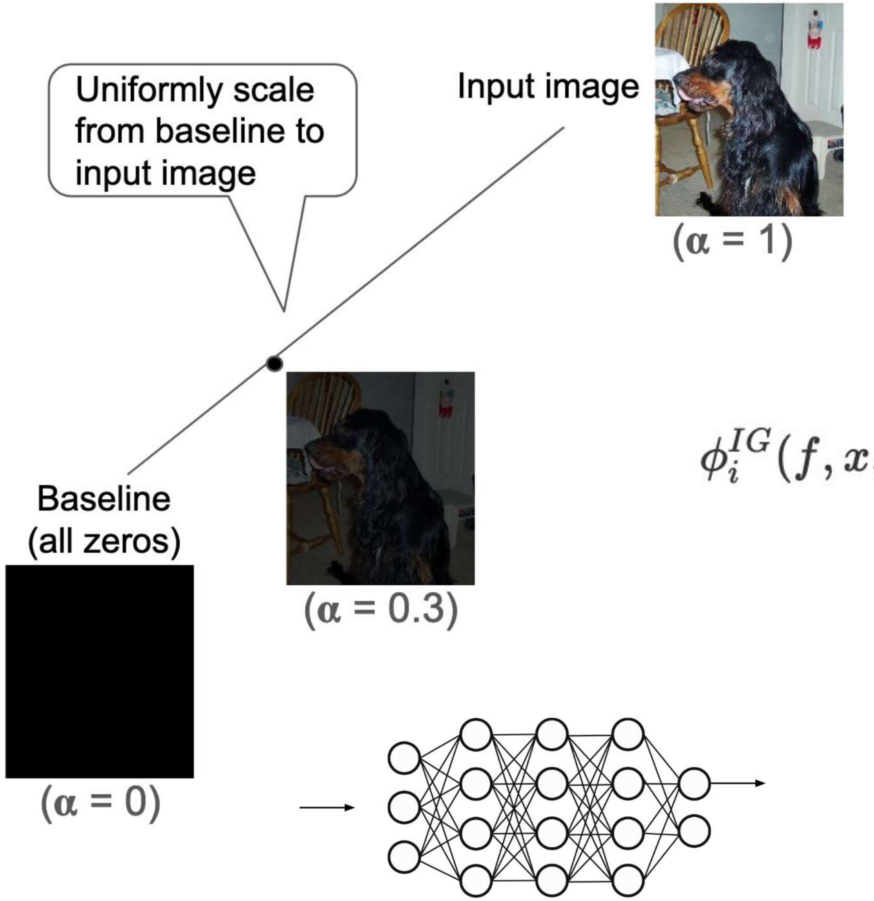
(α = 1)

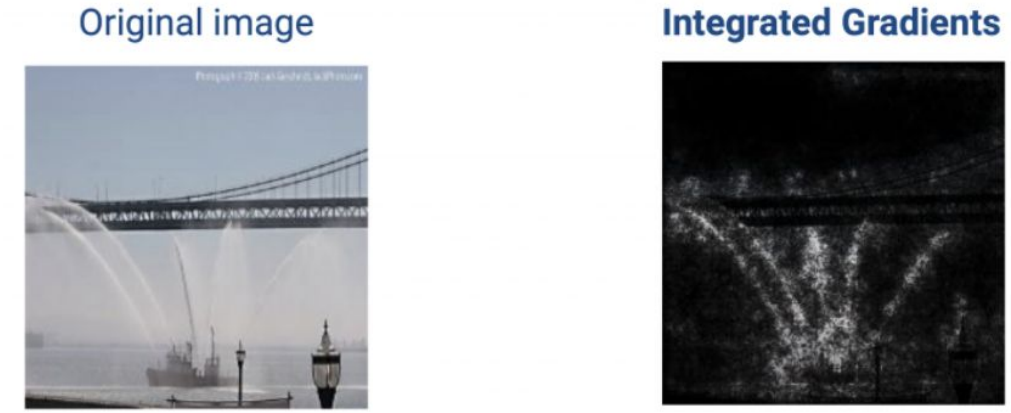
Baseline
(all zeros)

(α = 0)

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

Integrated gradients



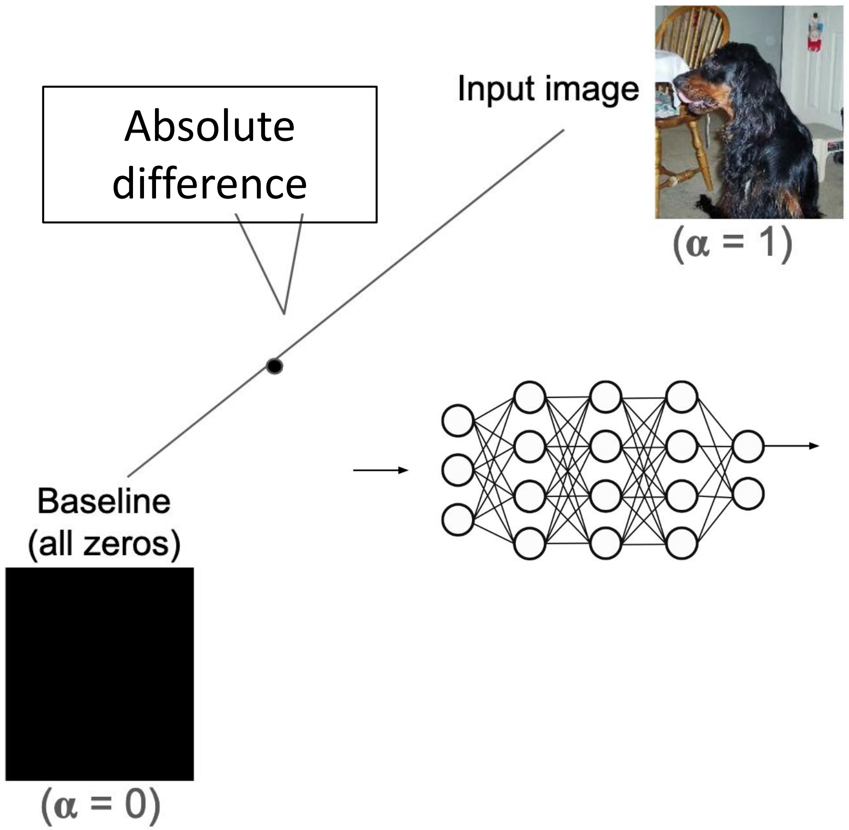


$$\phi_i^{IG}(f, x, x') = \overbrace{(x_i - x'_i)}^{\text{Difference from baseline}} \times \underbrace{\int_{\alpha=0}^{1}}_{\text{From baseline to input...}} \overbrace{\frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha}^{\text{...accumulate local gradients}}$$

Can we think of disadvantages of this method?

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

DeepLIFT

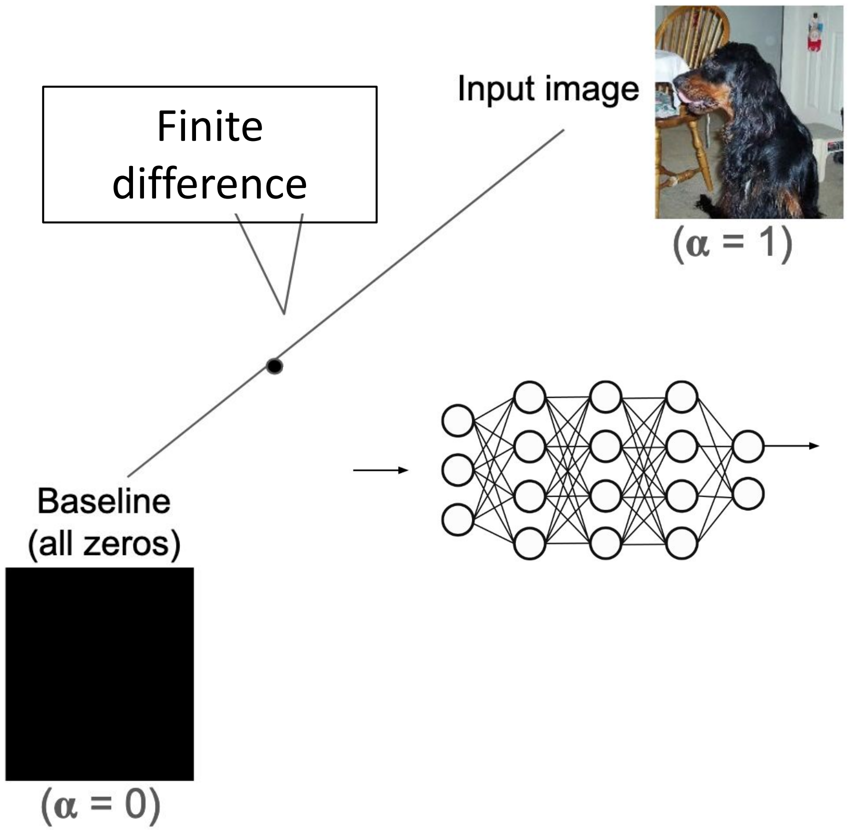


- Explain “difference from reference value” of output in terms of “difference from reference value” of inputs

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

Learning Important Features Through Propagating Activation Differences

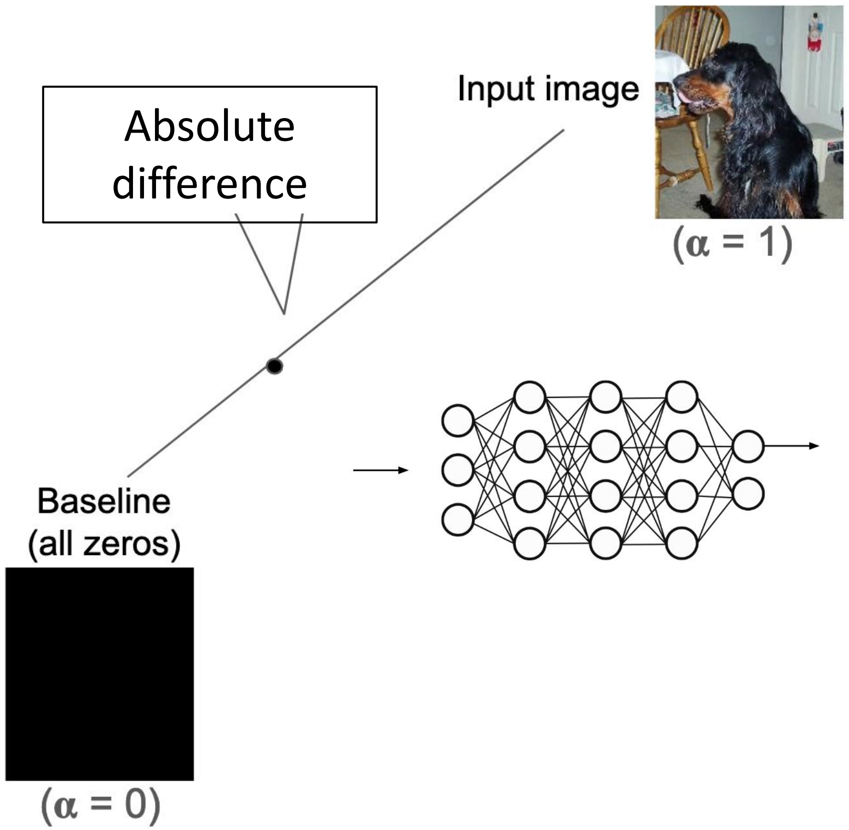
DeepLIFT



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

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

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

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

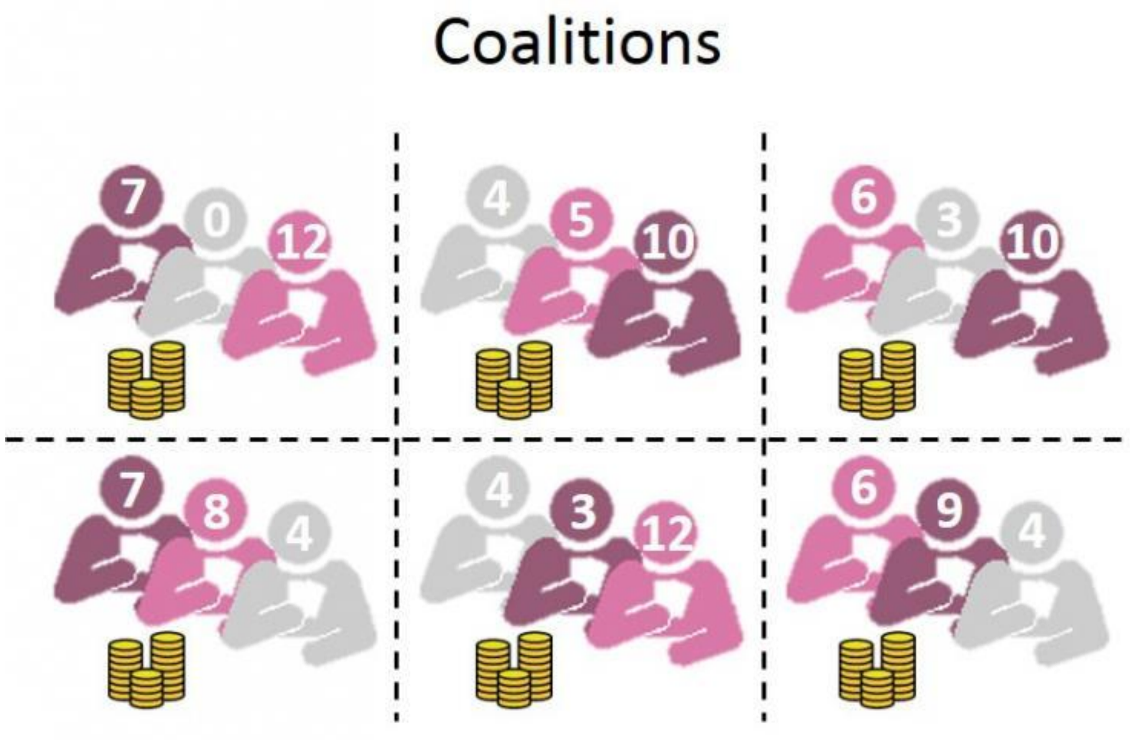
Slide courtesy: https://drive.google.com/file/d/0B15F_QN41VQXbkVkcTVQYTVQNVE/view

Learning Important Features Through Propagating Activation Differences

DeepLIFT (and Shapley values)

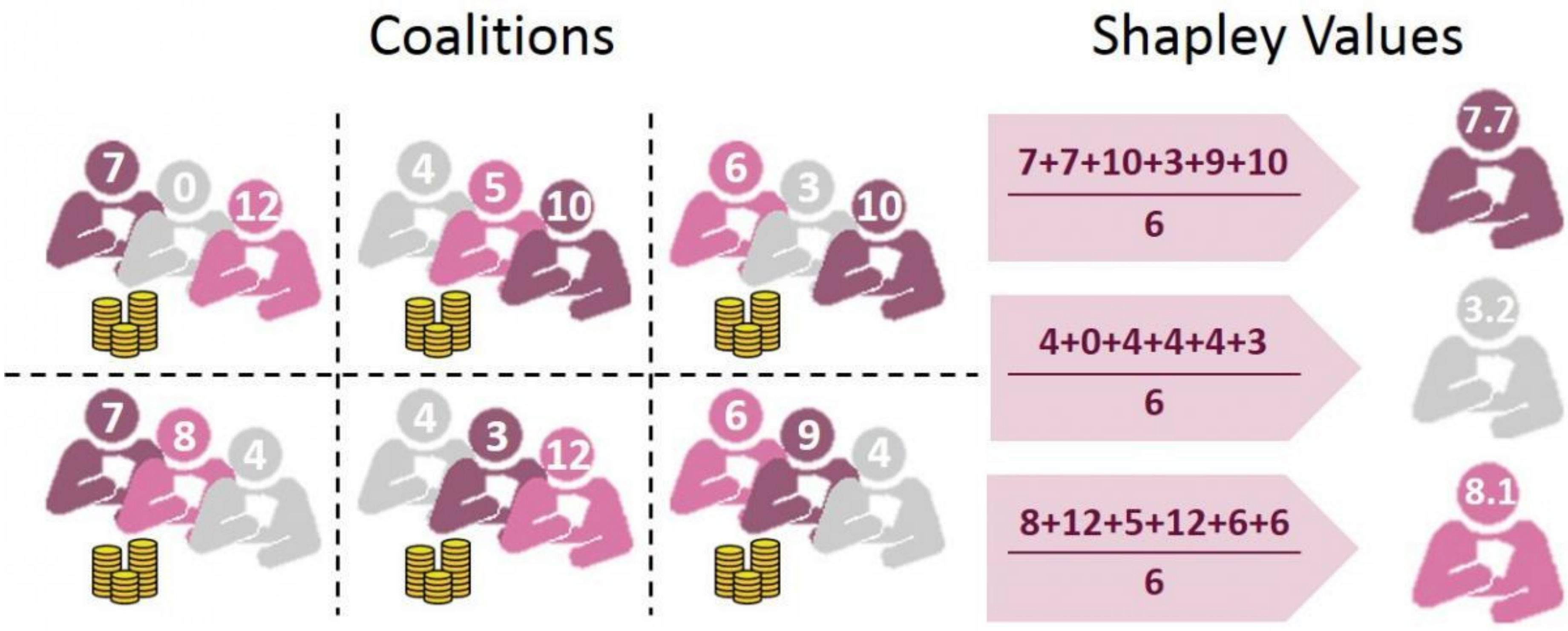
https://www.sia-partners.com/en/news-and-publications/from-our-experts/interpretable-machine-learning

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

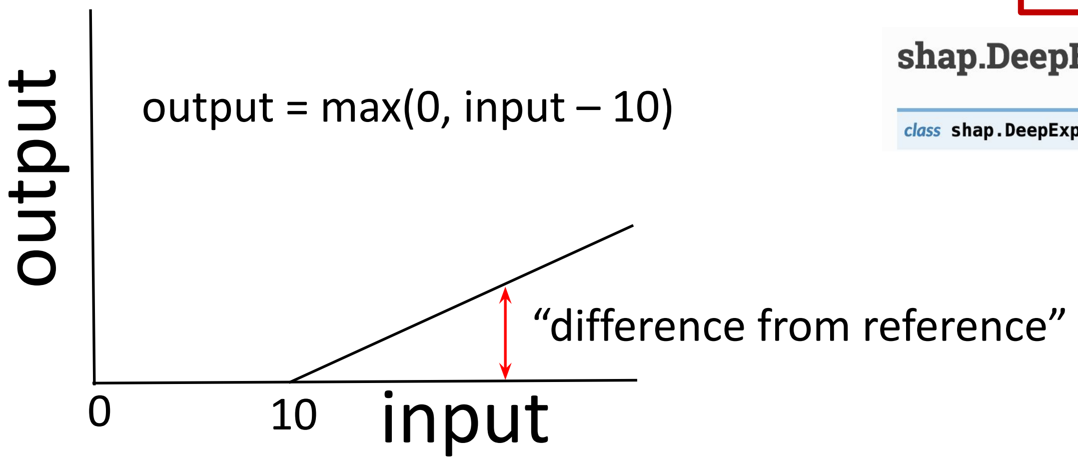
https://github.com/slundberg/shap

shap.DeepExplainer

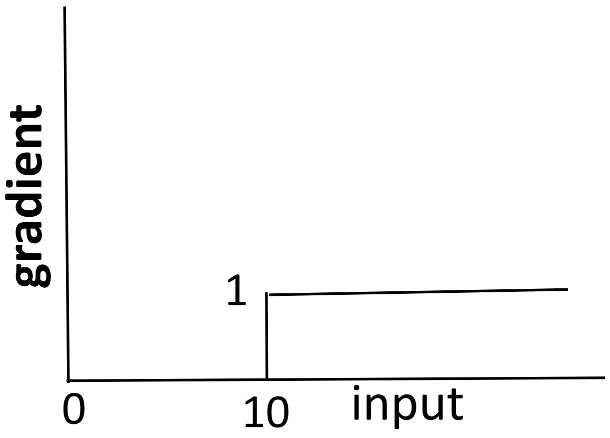
class `shap.DeepExplainer`(*model, data, session=None, learning_phase_flags=None*)

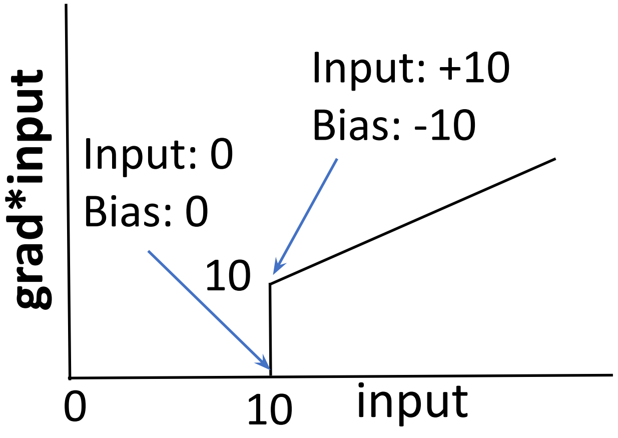
Solves thresholding problem

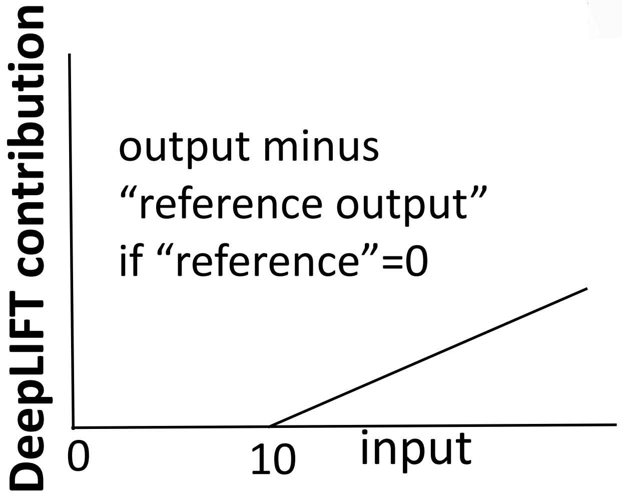
Can you think of any disadvantage of this method?



Any questions?







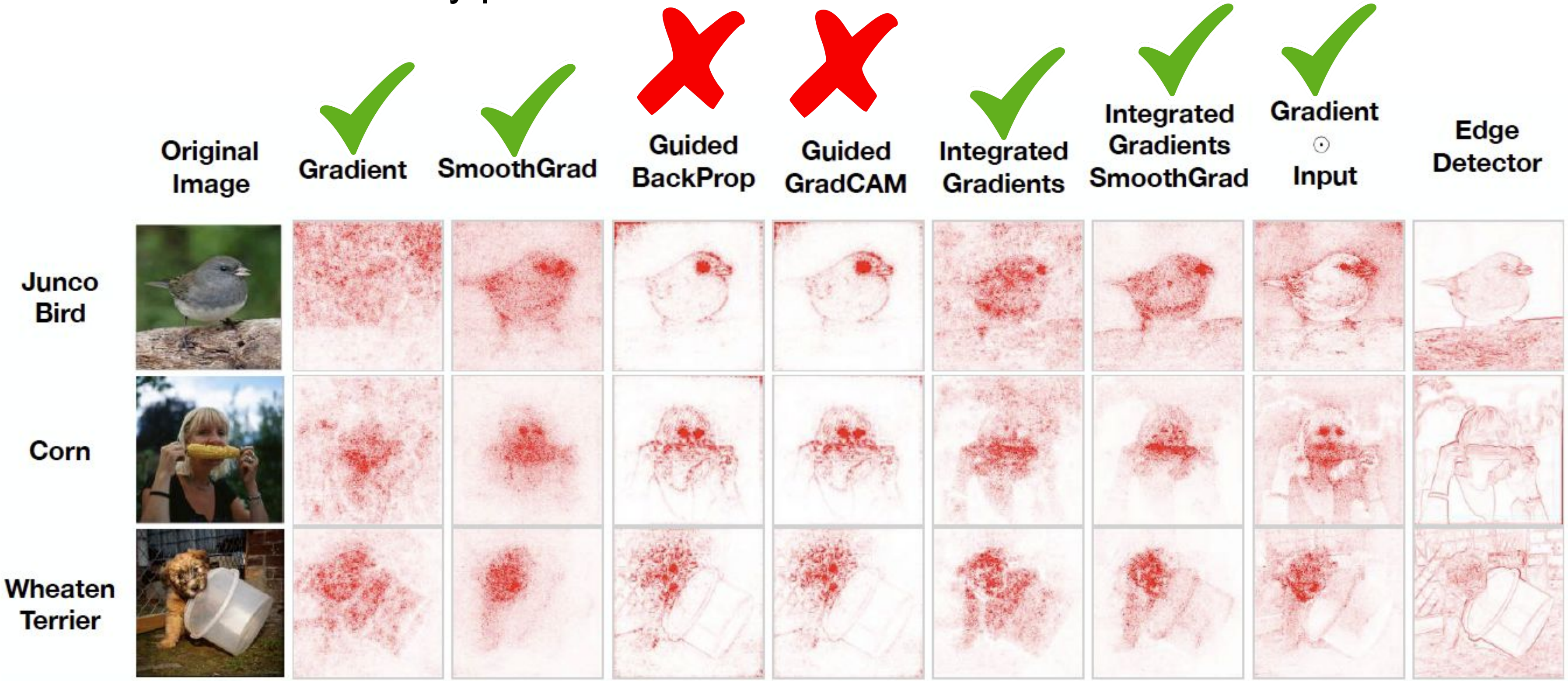
Slide courtesy: https://drive.google.com/file/d/0B15F_QN41VQXbkVkcTVQYTVQNVE/view

Testing different gradient-based methods

Model parameter randomization test compares the output of a saliency method on a trained model with the output of the saliency method on a randomly initialized untrained network of the same architecture.

Data randomization test compares a given saliency method applied to a model trained on a labeled data set with the method applied to the same model architecture but trained on a copy of the data set in which we randomly permuted all labels.

> "We find that reliance, solely, on visual assessment can be misleading.
>
> Through extensive experiments we show that some existing saliency methods are independent both of the model and of the data generating process."



Sanity Checks for Saliency Maps

Today's goal – learn about interpretation in DL

(1) Model architecture based methods

(2) Gradient-based methods

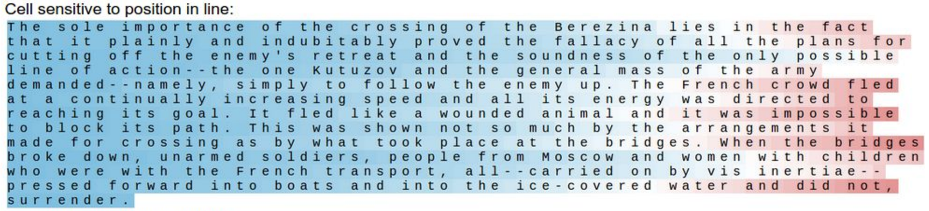
(3) Model agnostic methods

Next time!

Recap

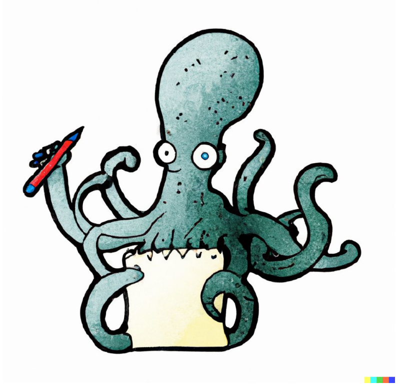
Model architecture based

- 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/