

# Interpretable and Explainable ML

Aramayis Dallakyan

Texas A&M University



June 17, 2022

# Material

The data and code is available

- [https://github.com/adallak/FAST2022\\_ExplainableAI](https://github.com/adallak/FAST2022_ExplainableAI)

# Motivation

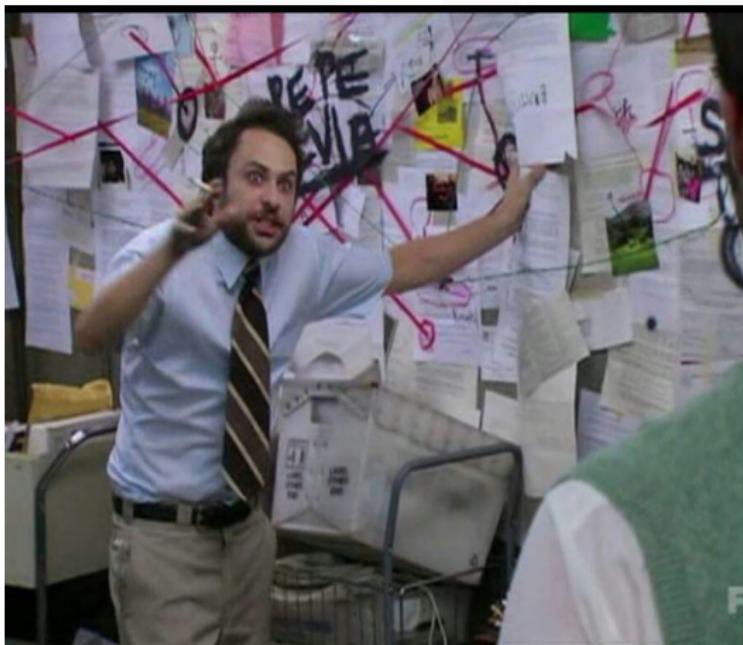
- Suppose you spend hours on training your model.
- You are satisfied with your evaluation metric.
- Are you done?



# Motivation

The goal of this workshop is to try to answer the following question:

**Why does my model predict what it predicts?**



# Overview

① Introduction to Explainable ML

② Model Specific Explanations

③ Model Agnostic

# Section 1

1 Introduction to Explainable ML

2 Model Specific Explanations

3 Model Agnostic

# Motivation

- One of the main problems in deploying Machine Learning algorithms is users trust to a prediction and a model.
- Both are highly affected based on the depth of human understanding of a model's behaviour, as opposed to perceiving it as a black box.
- Despite its widespread use, they treated and remain mostly as black boxes that do not explain their predictions in a way that practitioners can understand.

# Motivation

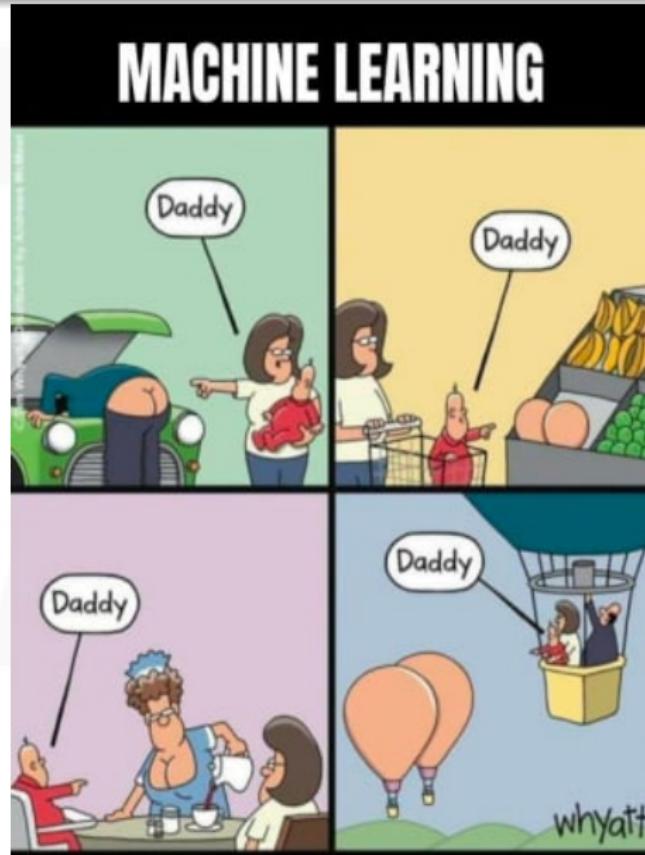
- ① Currently, most of the models are evaluated using performance metrics on a validation set.
- ② However, such metric may not be indicative since the real-world data is often different.
- ③ **The blind faith to the model** predictions can create discrimination and trust issues.



# Why it is important

- COMPAS is a proprietary model that is used in the U.S. Justice system for parole and bail decision.
- Recently it was accused on being racially biased, i.e., "This person is predicted to be arrested because they are black".
- Bank: model to predict who should get a loan or not.
- Computer Vision: Image Classification

# What Can Go wrong?



# Misuse Example

**Models are opinions embedded in mathematics.**

- Suppose you are hired at a big tech company to predict employees' performance.
- Available data shows the past performance reviews of individual employees for the past 10 years.
- What can go wrong if you apply black-box model?

# Misuse Example

**Models are opinions embedded in mathematics.**

- Suppose you are hired at a big tech company to predict employees' performance.
- Available data shows the past performance reviews of individual employees for the past 10 years.
- What can go wrong if you apply black-box model?
- What if that company tends to promote men more than women?

# Misuse Example

**Models are opinions embedded in mathematics.**

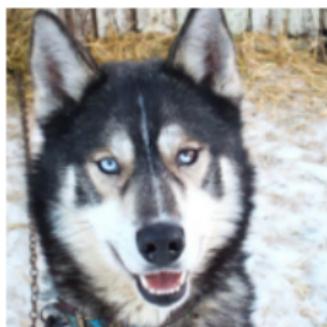
- Suppose you are hired at a big tech company to predict employees' performance.
- Available data shows the past performance reviews of individual employees for the past 10 years.
- What can go wrong if you apply black-box model?
- What if that company tends to promote men more than women?
- The model will learn the bias, and predict that men are more likely to be performant.

# Misuse Examples

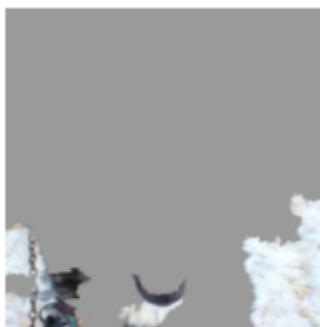
- Suppose now you work for a company whose main job is to classify wolves vs husky.
- The available data is pictures of wolves and huskies,
- What can go wrong if you apply black-box model [8]?

# Misuse Examples

- Suppose now you work for a company whose main job is to classify wolves vs husky.
- The available data is pictures of wolves and huskies,
- What can go wrong if you apply black-box model [8]?
- What if pictures of wolves show something different in the background?



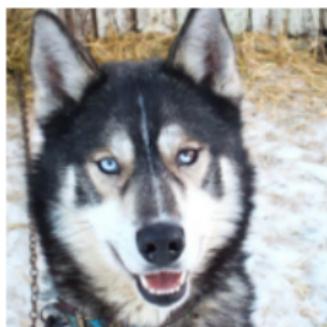
(a) Husky classified as wolf



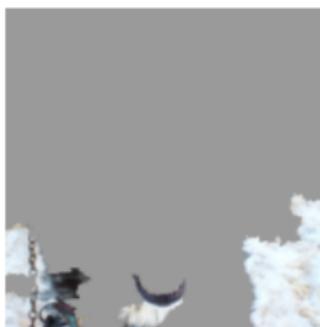
(b) Explanation

# Misuse Examples

- Suppose now you work for a company whose main job is to classify wolves vs husky.
- The available data is pictures of wolves and huskies,
- What can go wrong if you apply black-box model [8]?
- What if pictures of wolves show something different in the background?



(a) Husky classified as wolf



(b) Explanation

**Congratulations you just build a snow detector!!!**

# Misuse Example

A popular example from [2].



(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98



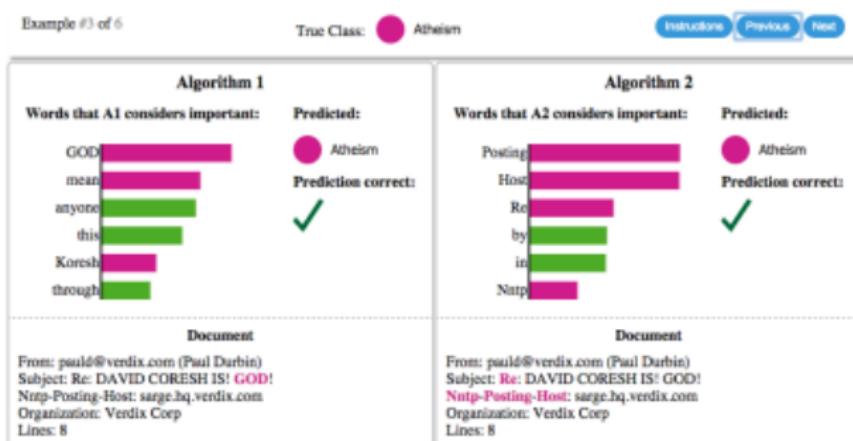
(B) **No Person: 0.99**, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97



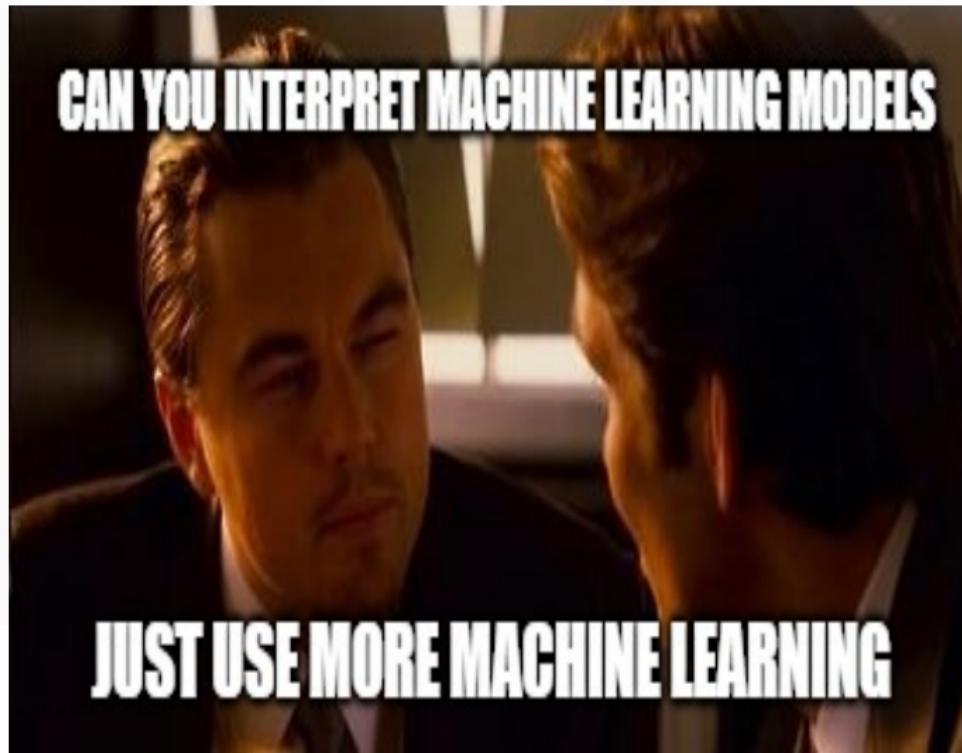
(C) **No Person: 0.97**,  
**Mammal: 0.96**, Water: 0.94,  
Beach: 0.94, Two: 0.94

# Misuse Example

- Now we want to predict whether a given document is about Atheism or Christianity.
- Algorithm 1 and Algorithm 2 have prediction accuracy 89%, 95%, respectively. Which one would you choose?



# Interpreting your prediction



# Interpretable vs Explainable ML

- There is a significant difference between interpretable and explainable models.
- A commonly used interpretable models are linear and logistic regression, decision trees, rule based methods and their extensions.

# Interpretability example

- Consider a linear regression

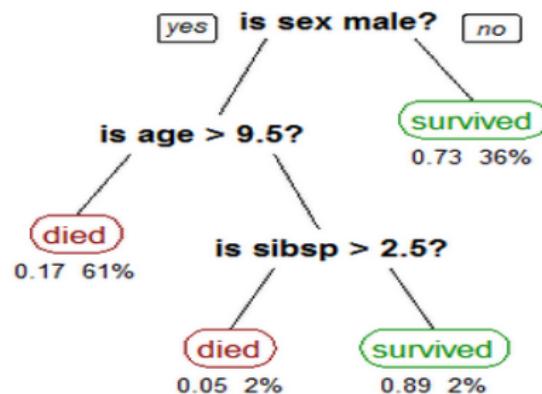
$$y = 5x - 0.6z + \epsilon$$

- or a logistic regression

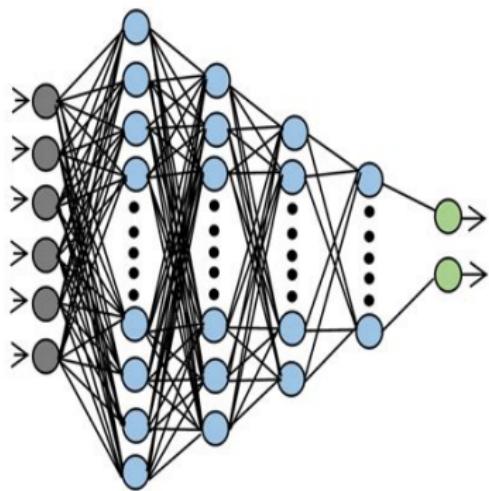
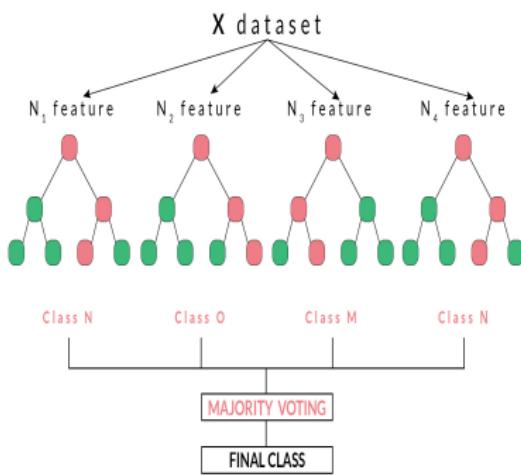
$$\log\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = 6x - 0.5z + \epsilon$$

and  $P(Y=1) = \frac{1}{1+e^{-(6x-0.5z)}}$

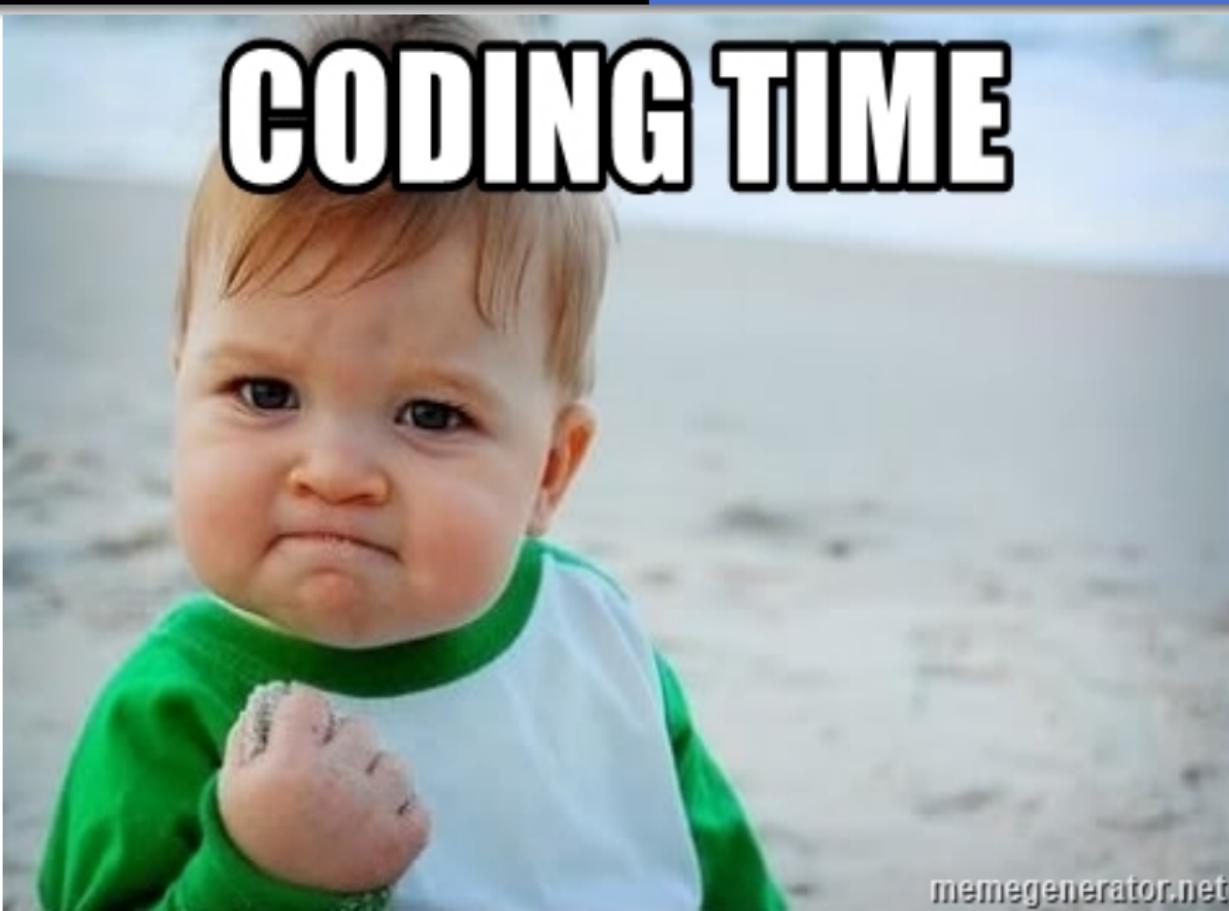
- or a decision tree



# Black Box Model



# CODING TIME

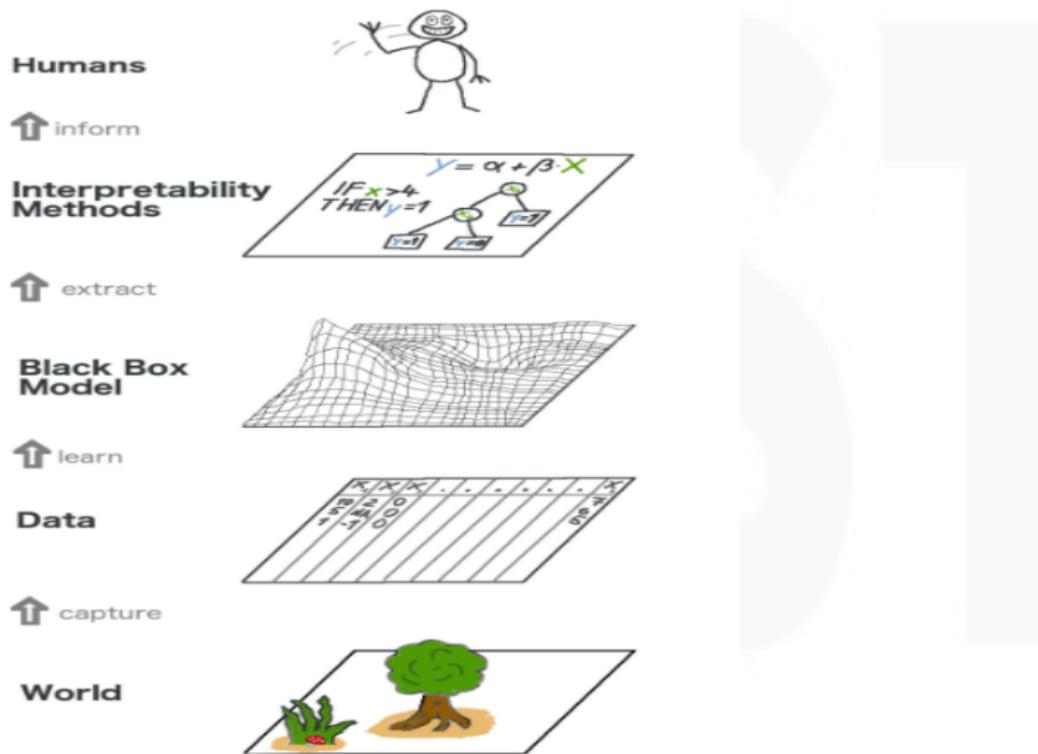


memegenerator.net

# Explainable ML

- In contrast to interpretable models, explainable methods do not create models that are inherently interpretable.
- But a post hoc model that explains the prediction of the original black box model.
- Do not use these methods for high stack decisions, but use as a tool for analysis and algorithmic audit.

# Explainable ML



Borrowed from [7]

# Explainable ML

- Rather than trying to create models that are inherently interpretable, Explainable ML creates a second(posthoc) model to explain the first black box model [10].
- Pros
  - Easy to construct
  - Model Agnostic: Treats the model as a black-box. Doesn't need to know how it makes predictions
- Cons
  - Can provide explanations that are not faithful to the original model.
  - Often do not provide enough details to understand what the black box is doing.

# Explainable Models

Explainable methods are separated into

- Model Specific: Grad-CAM, guided backpropagation, DeepLIFT, etc
- Model Agnostic: LIME, SHAP, surrogate trees, etc.

# Explainable ML

- Explanations are also come with two flavors:
  - **Global** describe the average behavior of a machine learning model.
    - Feature Importance,
  - **Local** methods explain individual predictions.
    - Why the loan for the customer X has been denied?

# Section 2

1 Introduction to Explainable ML

2 Model Specific Explanations

3 Model Agnostic

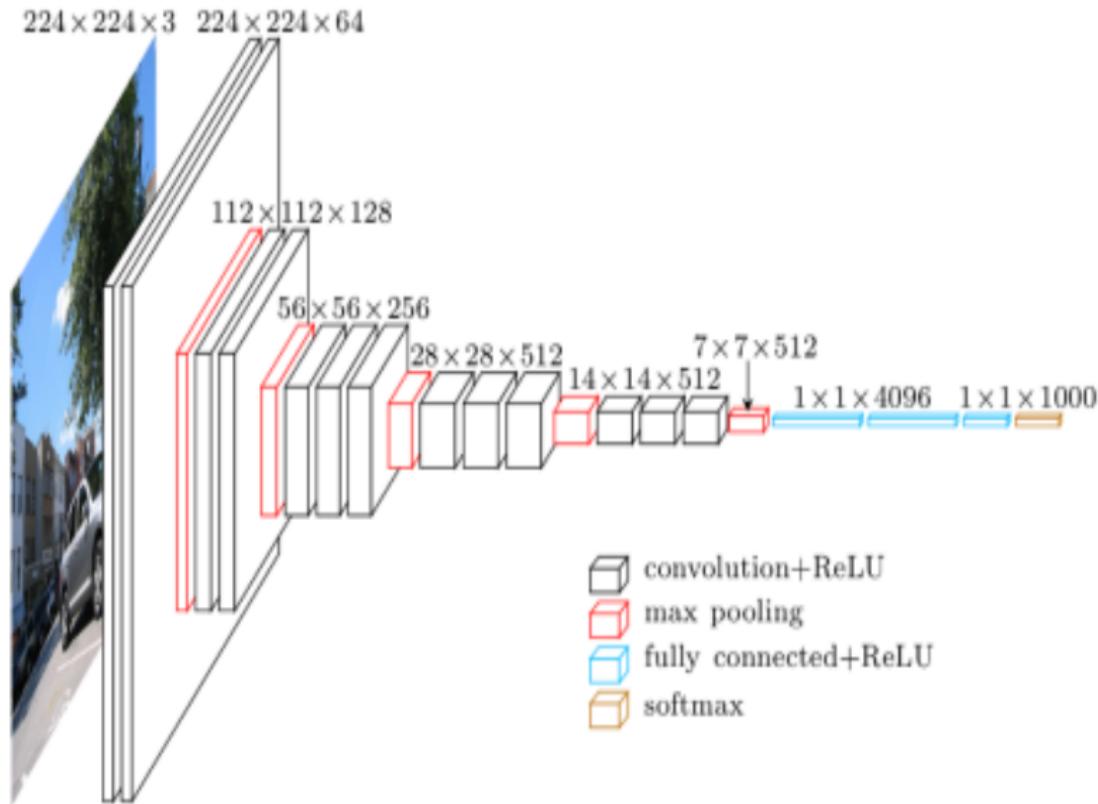
# Interpreting ConvNets

- As I mentioned before, deep learning models are "black boxes": they learn representation that are difficult to extract.
- Fortunately, this is not entirely true for the ConvNets.
- Convnets are highly amenable to visualization.

# Interpreting what ConvNets learn

- In this part of the workshop, our interest will be on interpreting what ConvNets learn.
- We will focus on two popular methods:
  - Grad-CAM
  - guided backpropagation

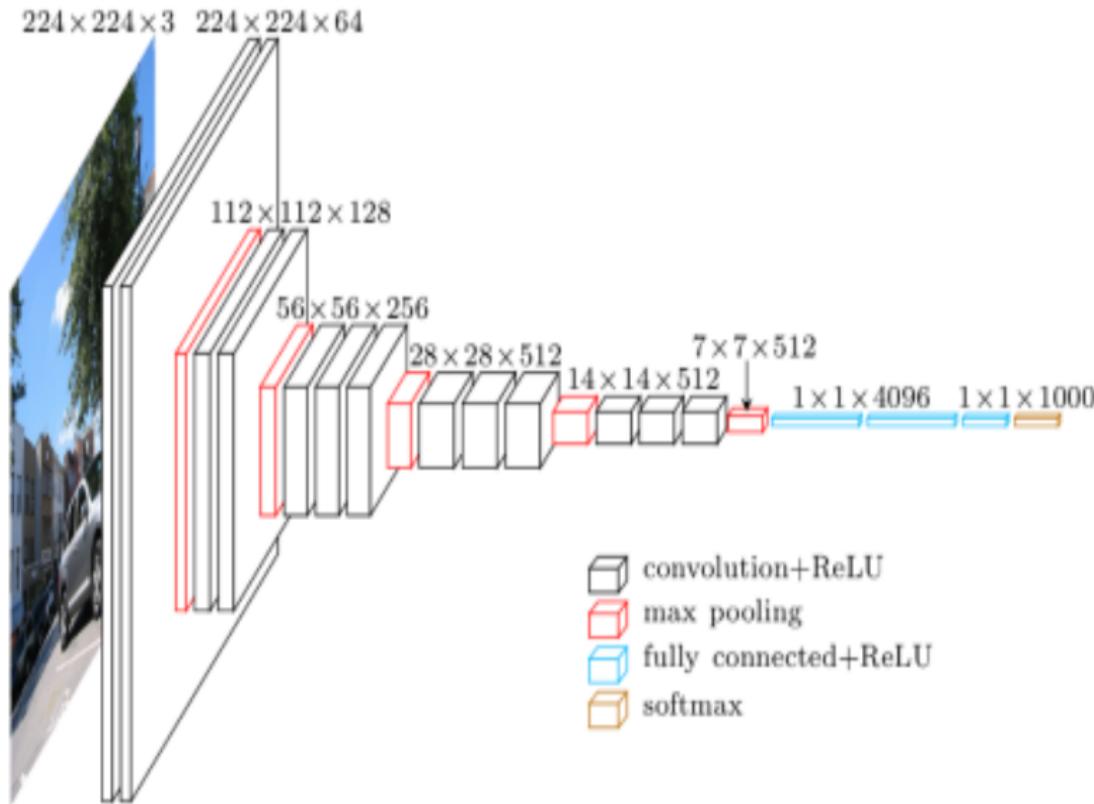
# Visualizing Convolutional Networks



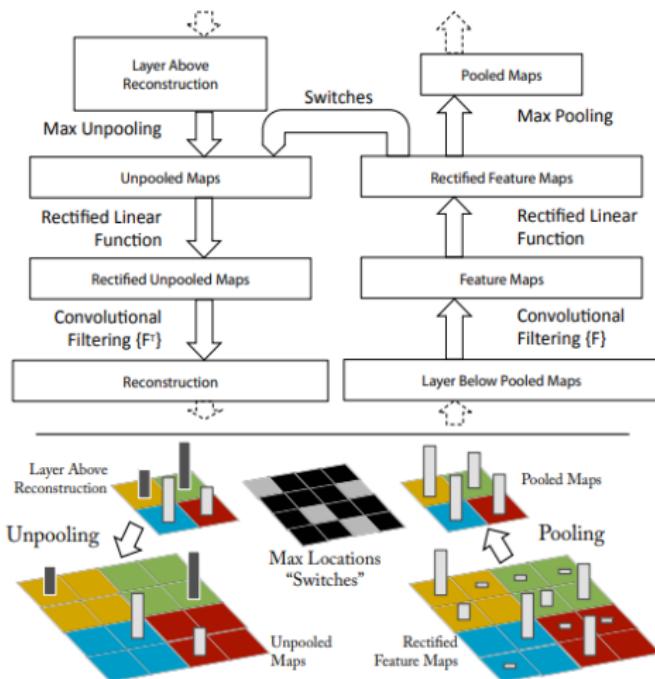
# Visualizing ConvNets

- Our goal is to display the values returned by various convolution and pooling layers, given a certain output.
- This tells us how an input is decomposed into the different filters learned by network.
- We want to visualize feature maps with three dimensions: width, height, and depth

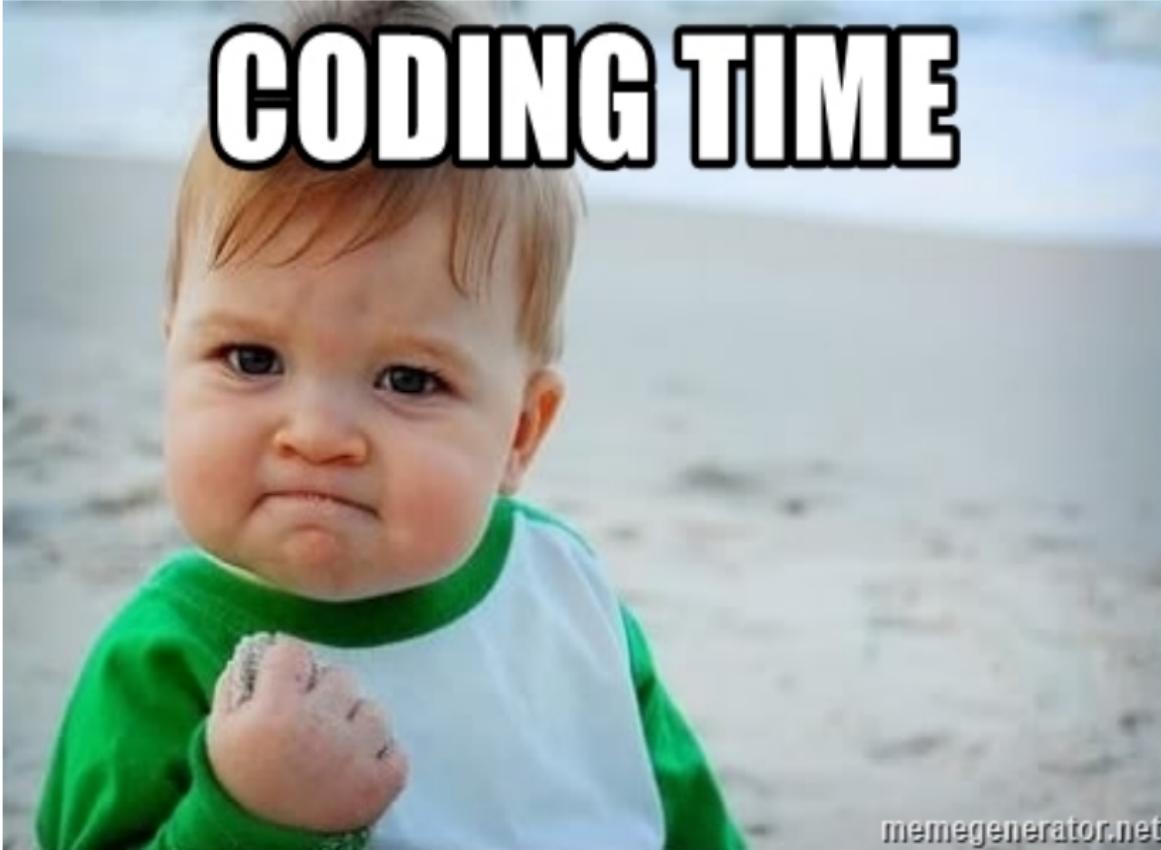
# Different Resolution of layers



# Deconvolution



Borrowed from [16]

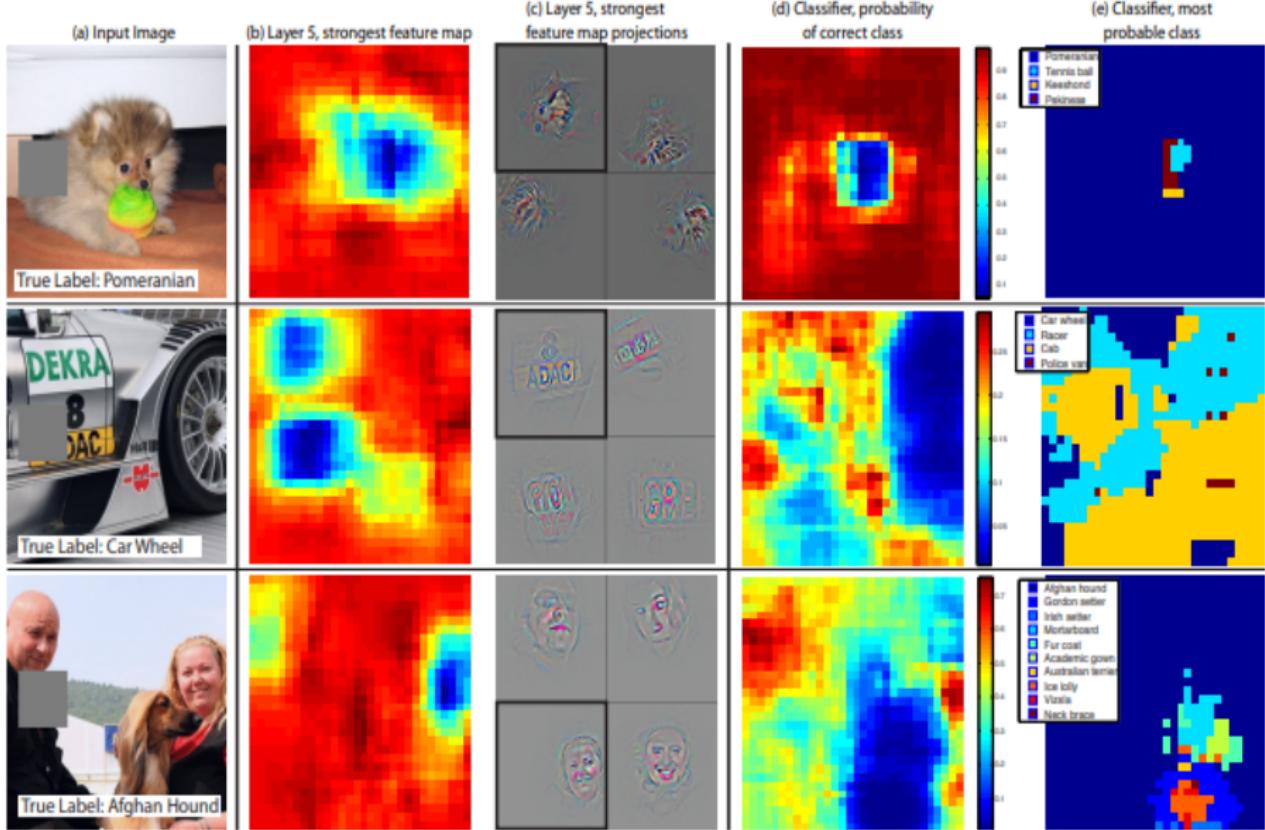


**CODING TIME**

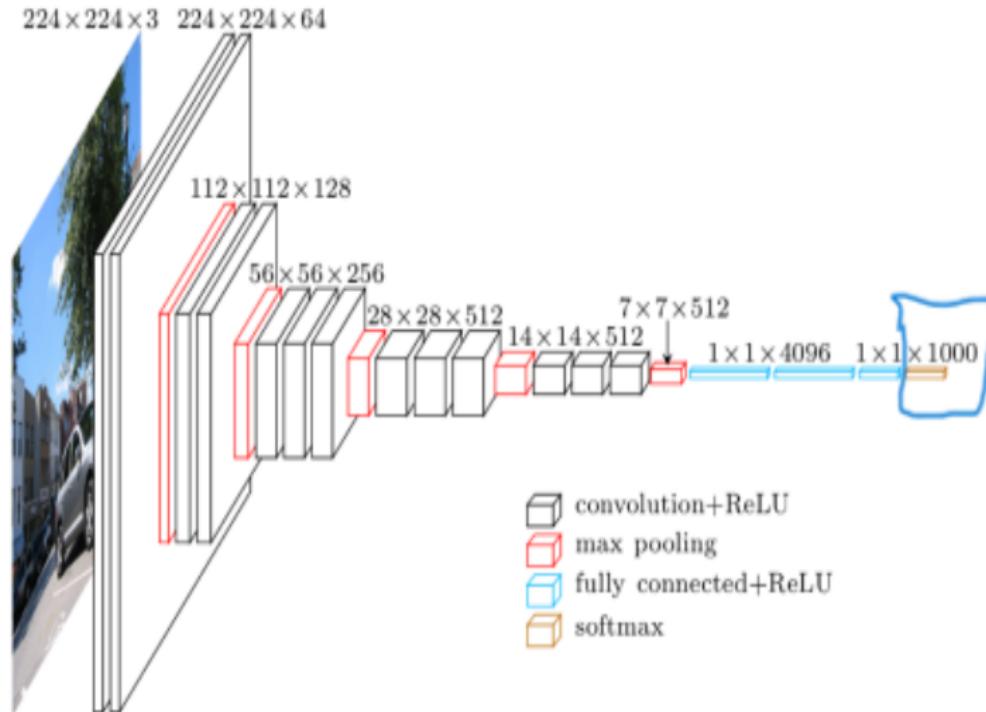
memegenerator.net

Details

# A Potential Good Project



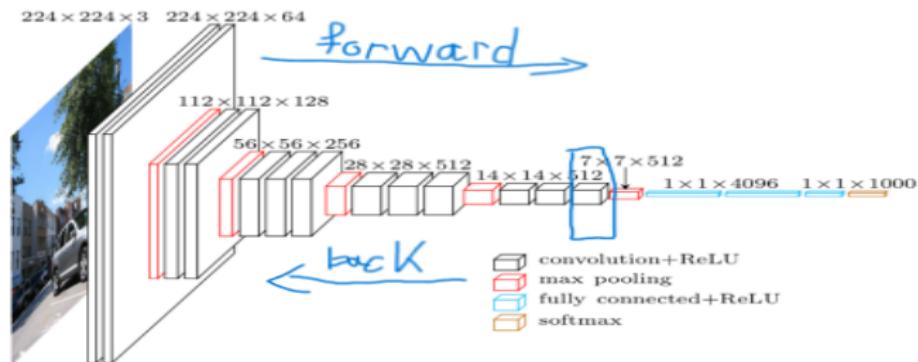
# Guided Backpropagation



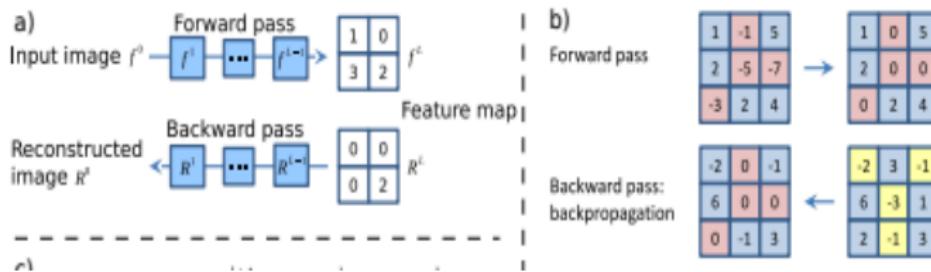
Question: Can we find an image that maximizes some class score?

# Visualize Gradient

A natural idea is to visualize the gradient of the particular layer.



Borrowed from [14]

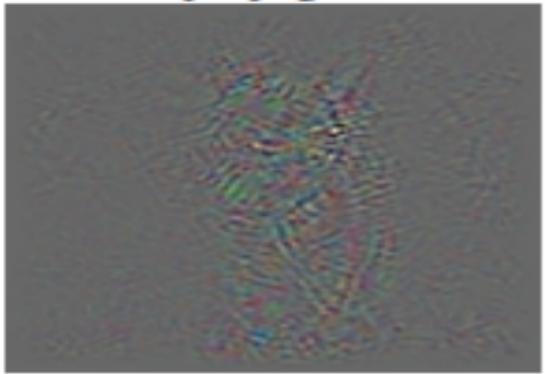


# Visualize Gradient

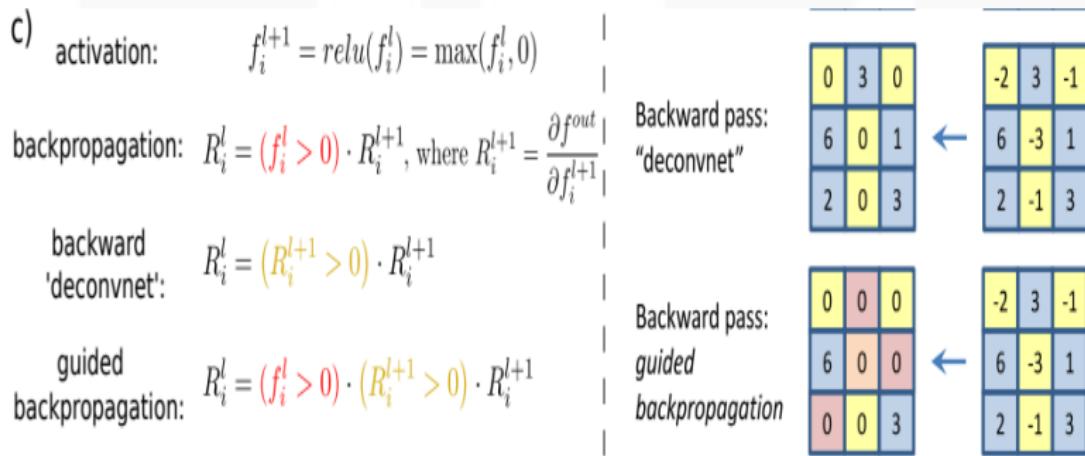


Borrowed from [14]

**backpropagation**



# Visualize Guided Backpropagation



[14]

# Visualize Gradient

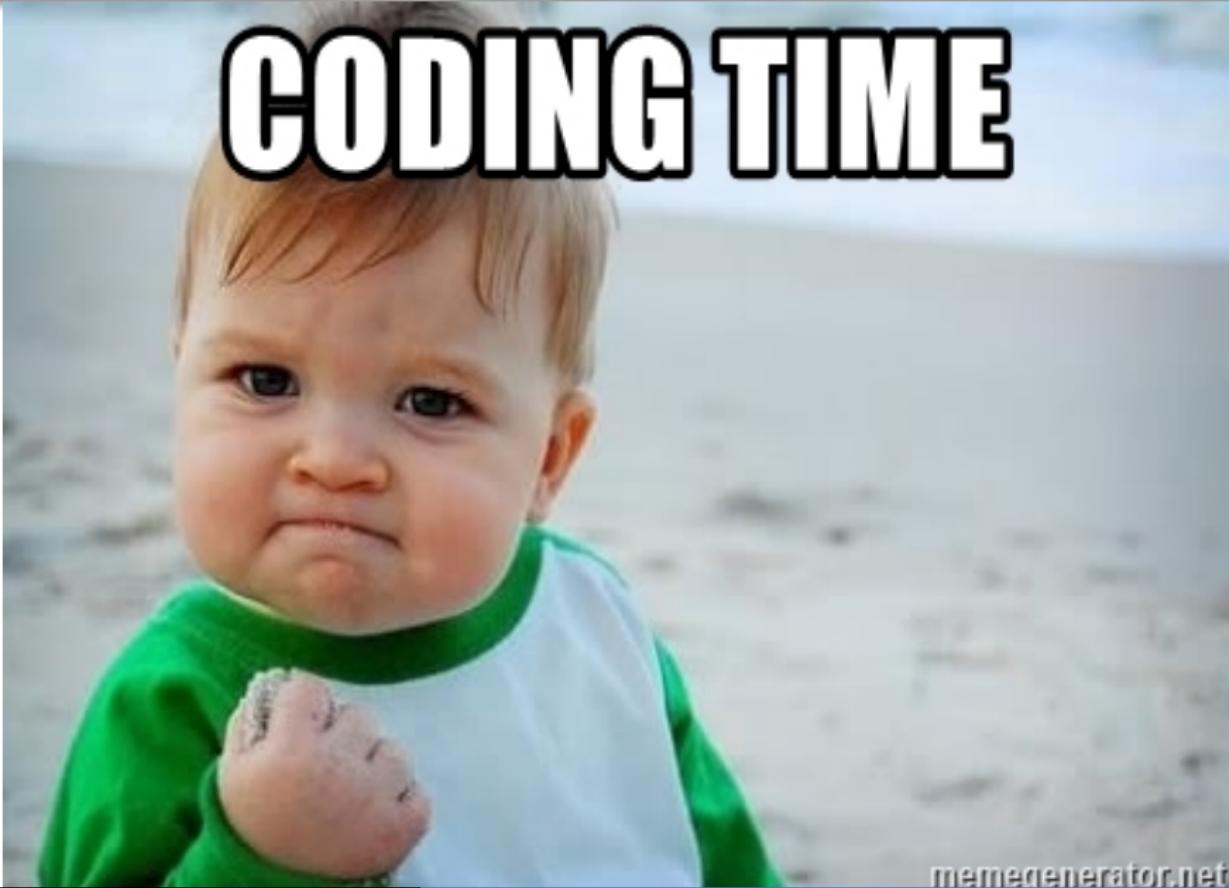


Borrowed from [14]

**guided backpropagation**



## Grad-CAM



**CODING TIME**

# Grad-CAM

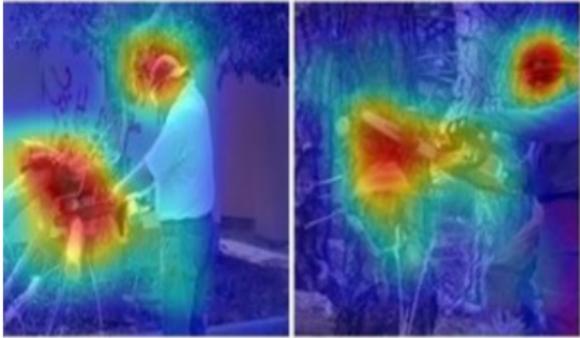
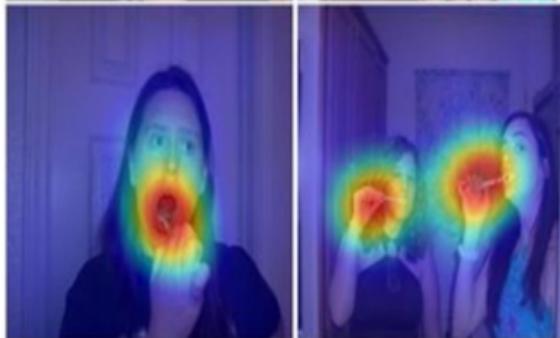
- We will now look at what are known as class discriminative saliency maps or class discriminative attribution maps.
- By saliency maps we mean maps or regions in an image that are salient for a given prediction.
- And by class discriminative saliency map, we mean a saliency map that helps distinguish one class from another.
- For example if we had a cat and a dog in an image which part of the image led to it being predicted to be a cat and which part of the same image led to it being predicted as a dog.

# Class Activation Maps

Brushing teeth



Cutting trees



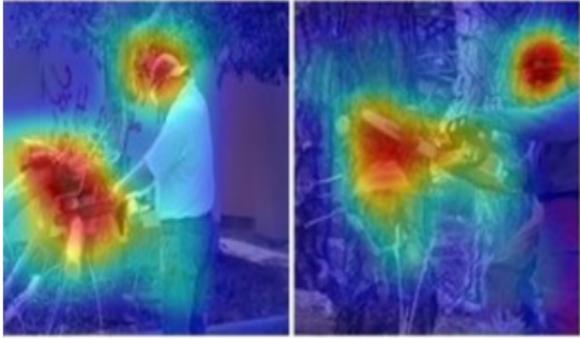
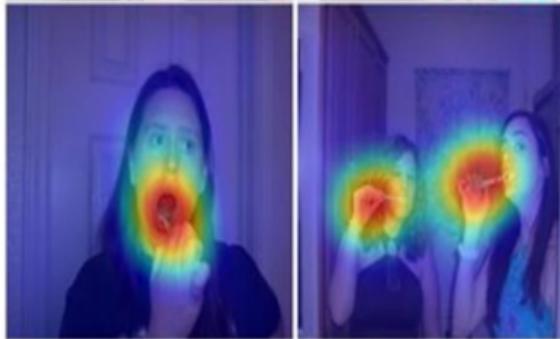
Borrowed from [17]

# Class Activation Maps

Brushing teeth

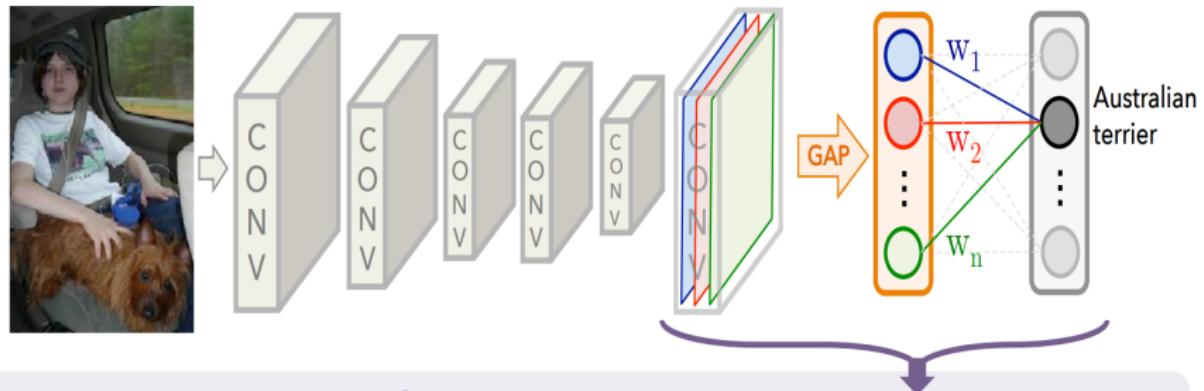


Cutting trees

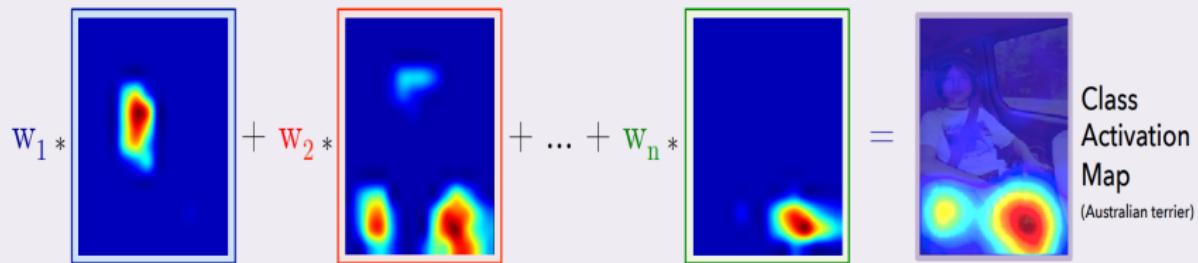


Borrowed from [17]

# Class Activation Maps

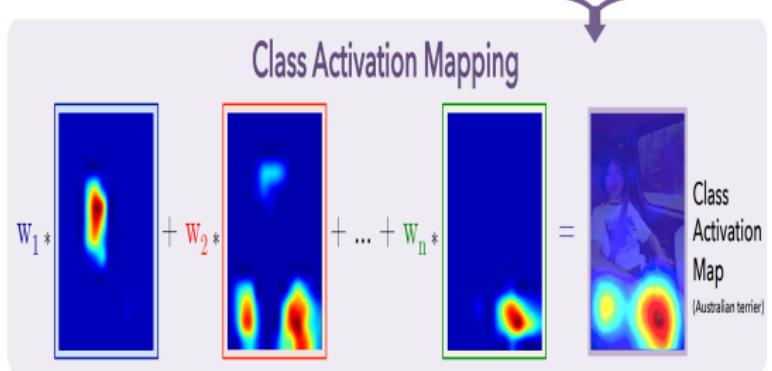
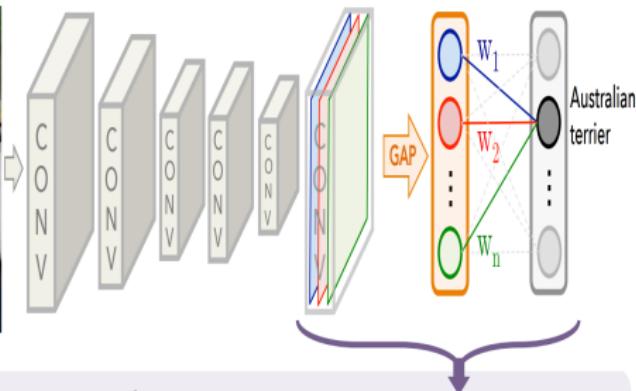


## Class Activation Mapping



Borrowed from [17]

# Class Activation Maps



$A_{ij}^k \rightarrow$  activation of unit  $k$  in the last convolutional layer.

$$\text{GAP: } F^k = \frac{1}{Z} \sum_{i,j} A_{ij}^k$$

$Y^c = \sum_k w_k^c F^k$ , input to the softmax for a given class and  $w_k^c$  is a weight corresponding to class  $c$  for unit  $k$ . It shows importance of  $F^k$  for class  $c$ .

$$P(Y = c) = \frac{\exp(Y^c)}{\sum_{c'} \exp(Y^{c'})}$$

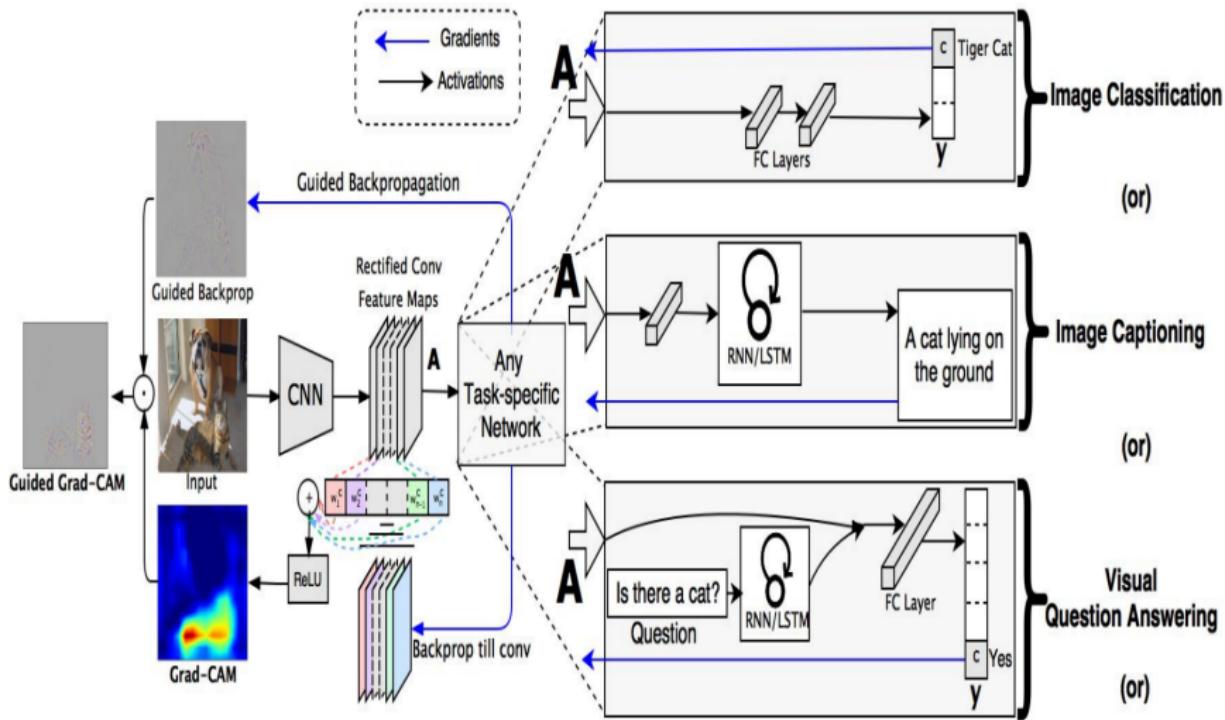
$$\begin{aligned} Y^c &= \sum_k w_k^c \frac{1}{Z} \sum_{i,j} A_{ij}^k \\ &= \frac{1}{Z} \sum_{i,j} \sum_k w_k^c A_{ij}^k \end{aligned}$$

$$\text{CAM} = \sum_k w_k^c A_{ij}^k$$

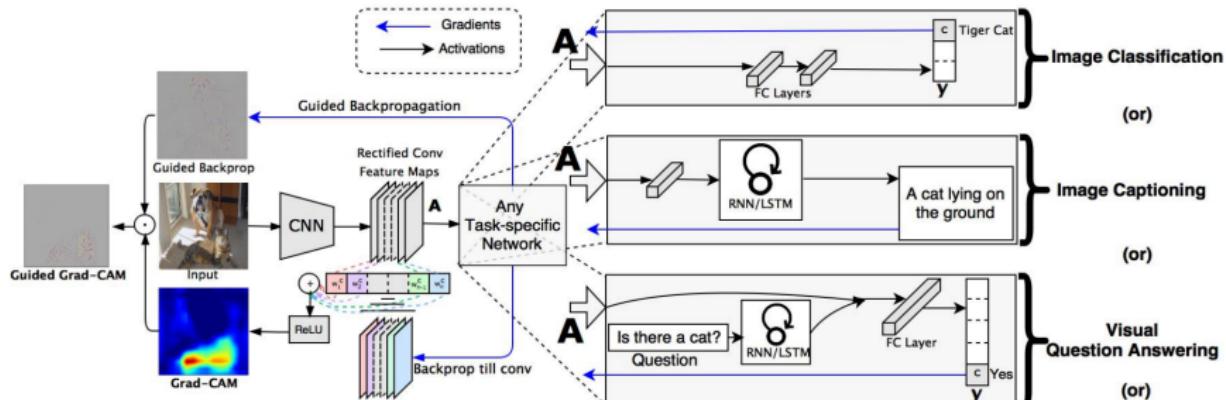
# CAM: Pros and Cons

- Advantages
  - It is class discriminative
  - Doesn't require a backward pass.
- Disadvantages
  - GAP imposes constraint on architecture
  - Need retraining to explained trained models.
  - Model may trade off accuracy for interpretability

# Grad-CAM



# Grad-CAM



From CAM  $Y^c = \sum_k w_k^c F^k$  and  $F^k = \frac{1}{Z} \sum_{i,j} A_{ij}^k$ , taking derivative

$$\frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A_{ij}^k}}{\frac{\partial F^k}{\partial A_{ij}^k}} \implies w_c^k = \frac{\partial Y^c}{\partial A_{ij}^k} Z$$

Summing both sides over  $i, j$

$$Zw_c^k = Z \sum_{i,j} \frac{\partial Y^c}{\partial A_{ij}^k}$$

Class features weights are gradients themselves. No retraining required.

# Grad-CAM: Methodology

- Uses gradients flowing from output class into activation maps of last convolutional layer as neuron importance weights

$$w_c^k = \sum_{i,j} \frac{\partial Y^c}{\partial A_{ij}^k}$$

- Similar to CAM

$$L_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k w_k^c A^k\right)$$

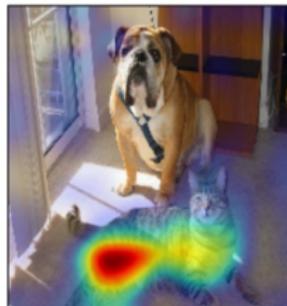
# Grad-CAM



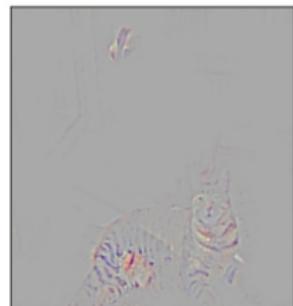
(a) Original Image



(b) Guided Backprop 'Cat'



(c) Grad-CAM 'Cat'



(d) Guided Grad-CAM 'Cat'



(g) Original Image



(h) Guided Backprop 'Dog'



(i) Grad-CAM 'Dog'

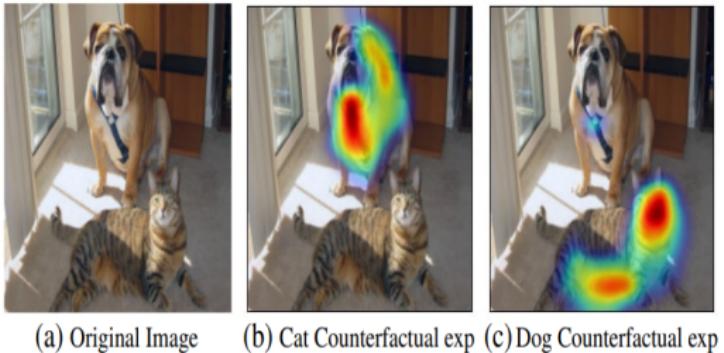


(j) Guided Grad-CAM 'Dog'

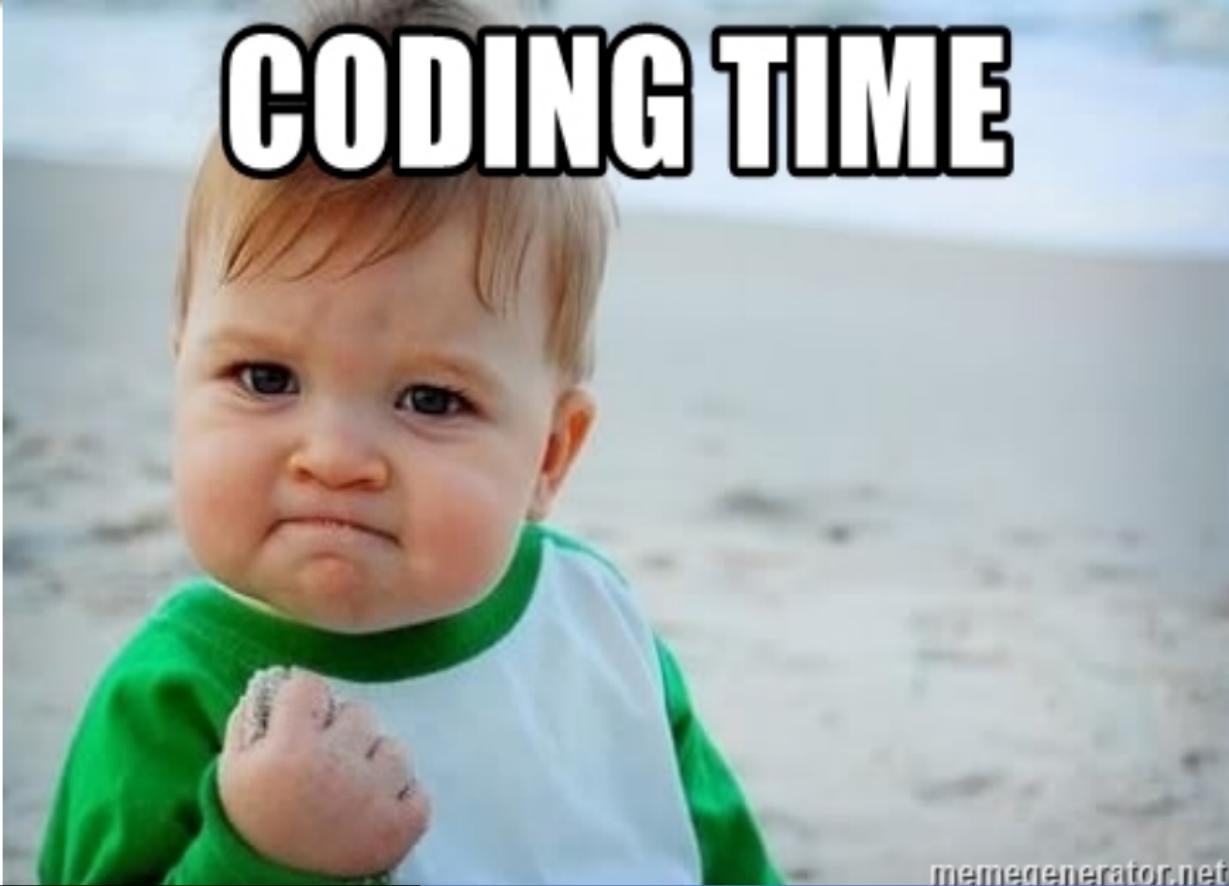
Interestingly, network predicts **Tiger Cat**, but Grad-CAM does not why. Guided Grad-CAM answers the question.

# Counterfactual Explanations

- By negating the value of gradients used in  $w_k^c$  we can visualize image patches that have adversarial affects.
- By removing or suppressing features occurring in such patches may improve model performance



## Grad-CAM



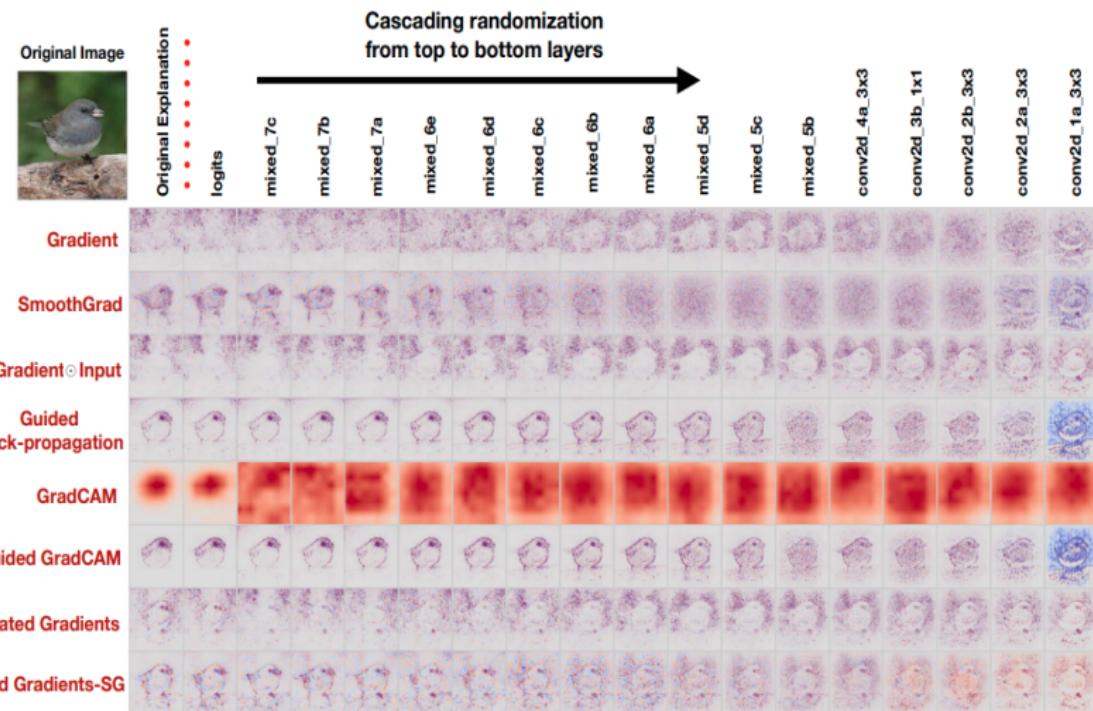
**CODING TIME**

# Recent Model Specific Methods

- Grad-CAM++ [4]: Improved version of Grad-CAM
- DeepLIFT [11]: Instead of gradient looks on how perturbation of inputs change the output.
- Integrated Gradient [15]: Gives different activation to gradients and then integrates over the results.
- SmoothGrad [13]: Adds gaussian noise to inputs and measures activation changes.
- XRAI [5]

# Warning on using Explainable Methods

One need to be extremely careful on visual interpretation of the explainable methods. See [1].



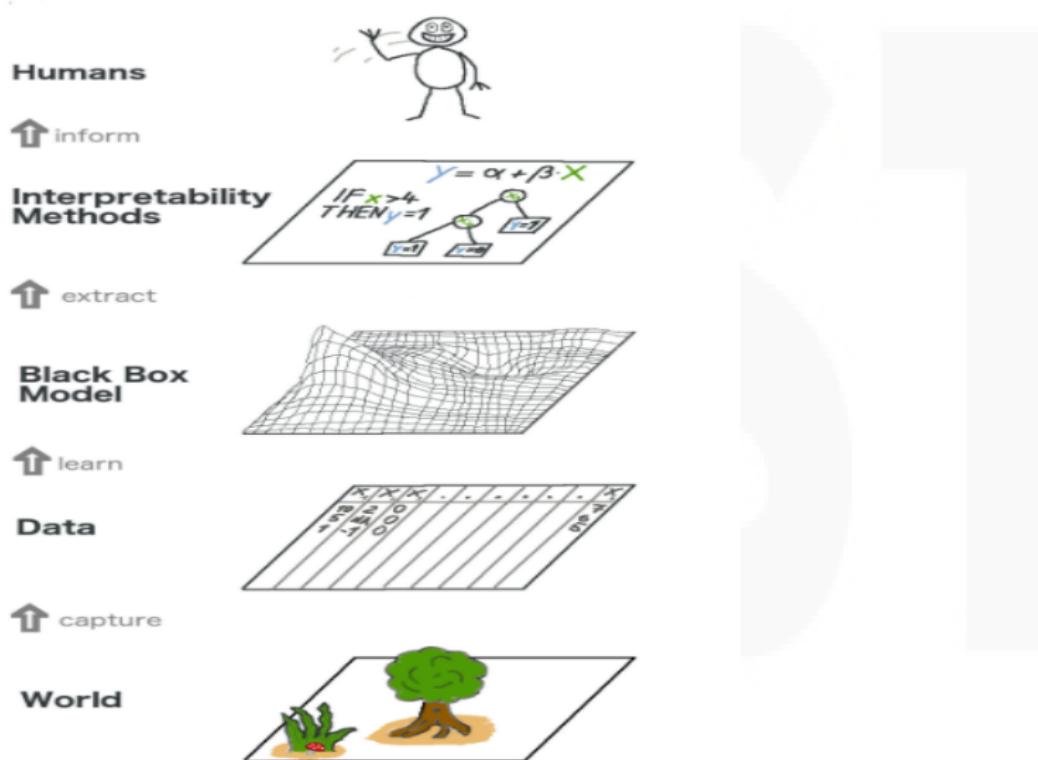
# Section 3

① Introduction to Explainable ML

② Model Specific Explanations

③ Model Agnostic

# Last Session Review



Borrowed from [7]

# Last Session Review

- Rather than trying to create models that are inherently interpretable, Explainable ML creates a second(posthoc) model to explain the first black box model [10].
- Pros
  - Easy to construct
  - Model Agnostic: Treats the model as a black-box. Doesn't need to know how it makes predictions
- Cons
  - Can provide explanations that are not faithful to the original model.
  - Often do not provide enough details to understand what the black box is doing.

# Last Session Review

Explainable methods are separated into

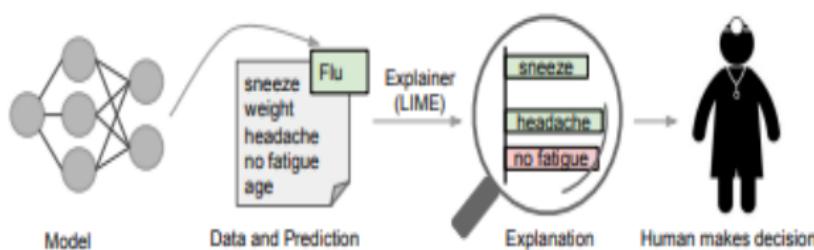
- Model Specific: Grad-CAM, guided backpropagation, DeepLIFT, etc
- Model Agnostic: LIME, SHAP, surrogate trees, etc.

# Last Session Review

- Explanations are also come with two flavors:
  - **Global** describe the average behavior of a machine learning model.
    - Feature Importance,
  - **Local** methods explain individual predictions.
    - Why the loan for the customer X has been denied?

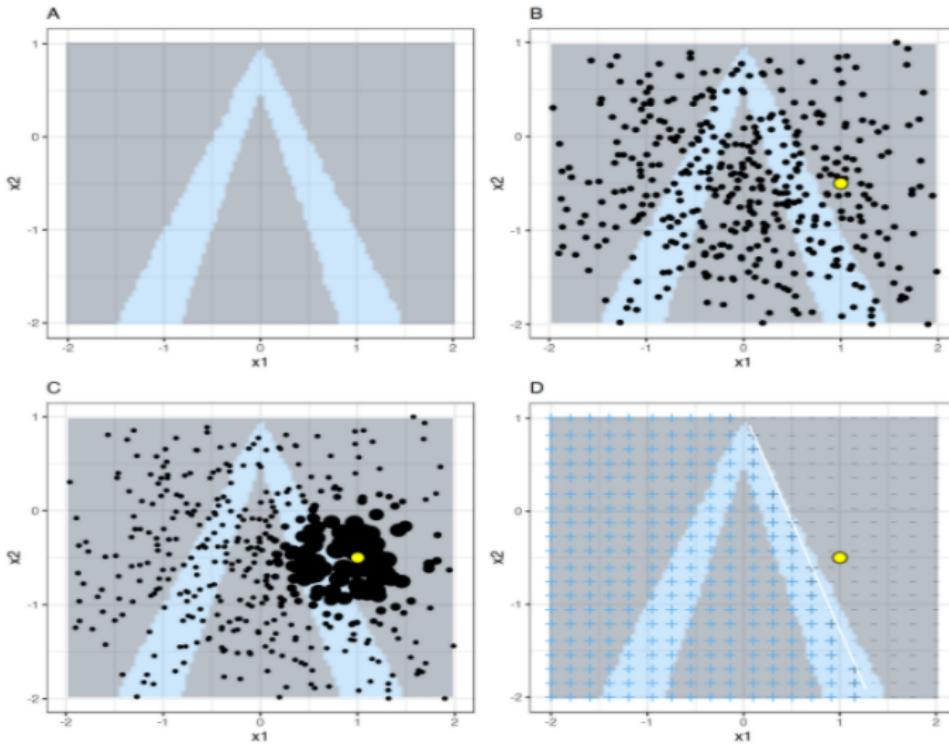
# LIME Algorithm

- Recall our goal is to understand why the machine learning model made a certain prediction.
- Local interpretable model-agnostic explanations (LIME) [9] focuses on training local surrogate models to explain individual predictions.



Borrowed from "Why Should I Trust You?: Explaining Prediction for Any Classifier" [9]

# How does it work?



# A little bit details

- The generation of new samples is a little bit tricky, let's go over it using a toy example.
- Suppose FAST wants to predict whether a student in a given batch will pass the final exam using previous observations.

Student	GPA	Workshop	passed
St1	3.3	0	0
St2	3.65	1	1
St3	3.7	0	0
St4	3.4	0	1
St5	3.45	1	0

- Suppose FAST used some fancy machine learning algorithm and wants to explain prediction for St5, we denote this observation by  $x_*$ .

# A little bit details

One way to generate new 100 samples is following:

- For  $i = 1 \dots 100$  do:
- GPA is continuous, we will find mean  $\mu = 3.5$  and st.dev  $\sigma = 0.15$  then new sample point  $x_{i1} = \mu + z\sigma$ , where  $z = N(0, 1)$ .
- Workshop is categorical and probability of 1 is  $2/5$ , thus new observations is  $x_{i2} = \text{Bernoulli}(2/5)$ .
- then  $x = [x_{i1}, x_{i2}]$  and the closeness is defined by

$$\pi_{x_*}(x) = \exp\left(-\frac{\|x_* - x\|_2^2}{\tau^2}\right)$$

If the original dataset is high dimensional let say  $x \in R^p$ , then first we create lower dimensional representation by randomly selecting  $K$  of them and implemented the above procedure for the selected variables.

# A little bit Math

- Recall that we want to find a model that locally approximates a black-box model  $f(\cdot)$  around the instance of interest  $x_*$
- We consider a class of  $G$  simple models, for example linear models  $g(\tilde{X}) = \tilde{X}\beta$
- To find the approximation, we minimize

$$\hat{\beta} = \arg \min_{\beta} L(f, g, \pi(x_*)) + \lambda \|\beta\|,$$

where  $\pi_x(x_*)$  defines a neighborhood of  $x_*$  and  $L(\cdot)$  is a loss function. **What is the most popular loss function?**

# A little bit Math

- Recall that we want to find a model that locally approximates a black-box model  $f(\cdot)$  around the instance of interest  $x_*$
- We consider a class of  $G$  simple models, for example linear models  $g(\tilde{X}) = \tilde{X}\beta$
- To find the approximation, we minimize

$$\hat{\beta} = \arg \min_{\beta} L(f, g, \pi(x_*)) + \lambda \|\beta\|,$$

where  $\pi_x(x_*)$  defines a neighborhood of  $x_*$  and  $L(\cdot)$  is a loss function. **What is the most popular loss function?**

- In practice we choose  $K$  feature and there is no need for  $\|\beta\|$ .

$$\hat{\beta} = \arg \min_{\beta} \sum_{z, z'} \pi_x(z) (f(z) - g(z'))^2$$

# LIME algorithm

**Require:**  $x^*$ - observation to be explained

**Require:** sample size -  $N$ , complexity -  $K$

Let  $x' = h(x^*)$  lower-dimensional representation

**for**  $i = 1 \dots N$  **do**

$z'[i] = \text{sample\_around}(x')$

$z[i] = h(z'[i])$  recover obervation to original space

$y[i] = f(z[i])$  prediction for the new observation

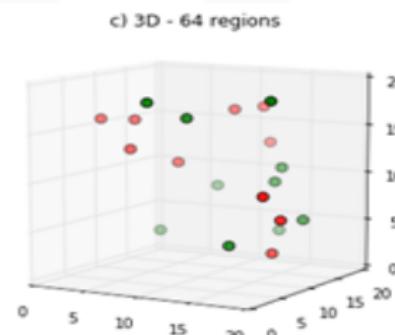
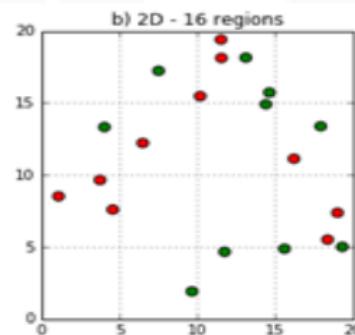
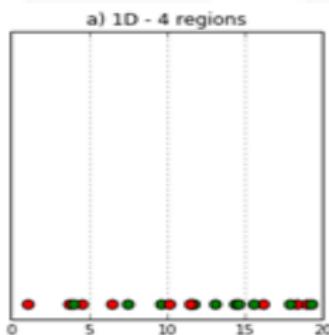
$\pi[i] = \pi_x(z)$  estimate similarity

**end for**

**return**  $\hat{\beta} = \arg \min_{\beta} \sum_{z,z'} \pi_x(z) (f(z) - g(z'))^2$

# More Details

- Why do we need a lower-dimensional representation?
- Consider an image with  $244 \times 244$  pixels, i.e. the original space is  $3 \times 244 \times 244$  and input space is 178,608-dimensional.
- Recall for the LIME we need to sample around  $x$  which is a hard task for high dimensional problems (**Curse of Dimensionality**)



# More Details

- To avoid curse of dimensionality the space can be transformed into superpixels, which are treated as binary features that can be turned on or off.



1

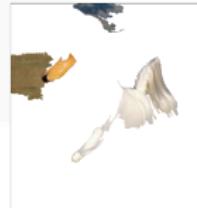
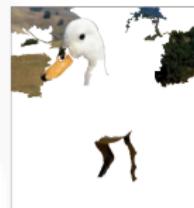
- Thus in this case the black-box model  $f()$  operates on 178,608-dimensional space and  $g()$  in  $\{0, 1\}^{100}$

---

<sup>1</sup>Picture is borrowed from [3]

# Sampling around the instance

- Due to high-dimension usually we cannot sample from the original data, which is sparse.
- In practice, we create new data points by perturbing the instance we want to explain.
- 



2

- 

<sup>2</sup>Picture is borrowed from [3]

# Explanation

- Central plot shows importance of the top features for this prediction
  - Value corresponds to the weight of the linear model
  - Direction corresponds to whether it pushes in one way or the other
- Numerical features are discretized into bins
  - Easier to interpret: weight == contribution / sign of weight == direction



# LIME for Images



$P($   $) = 0.54$



$P($   $) = 0.07$



$P($   $) = 0.05$



# Some drawbacks

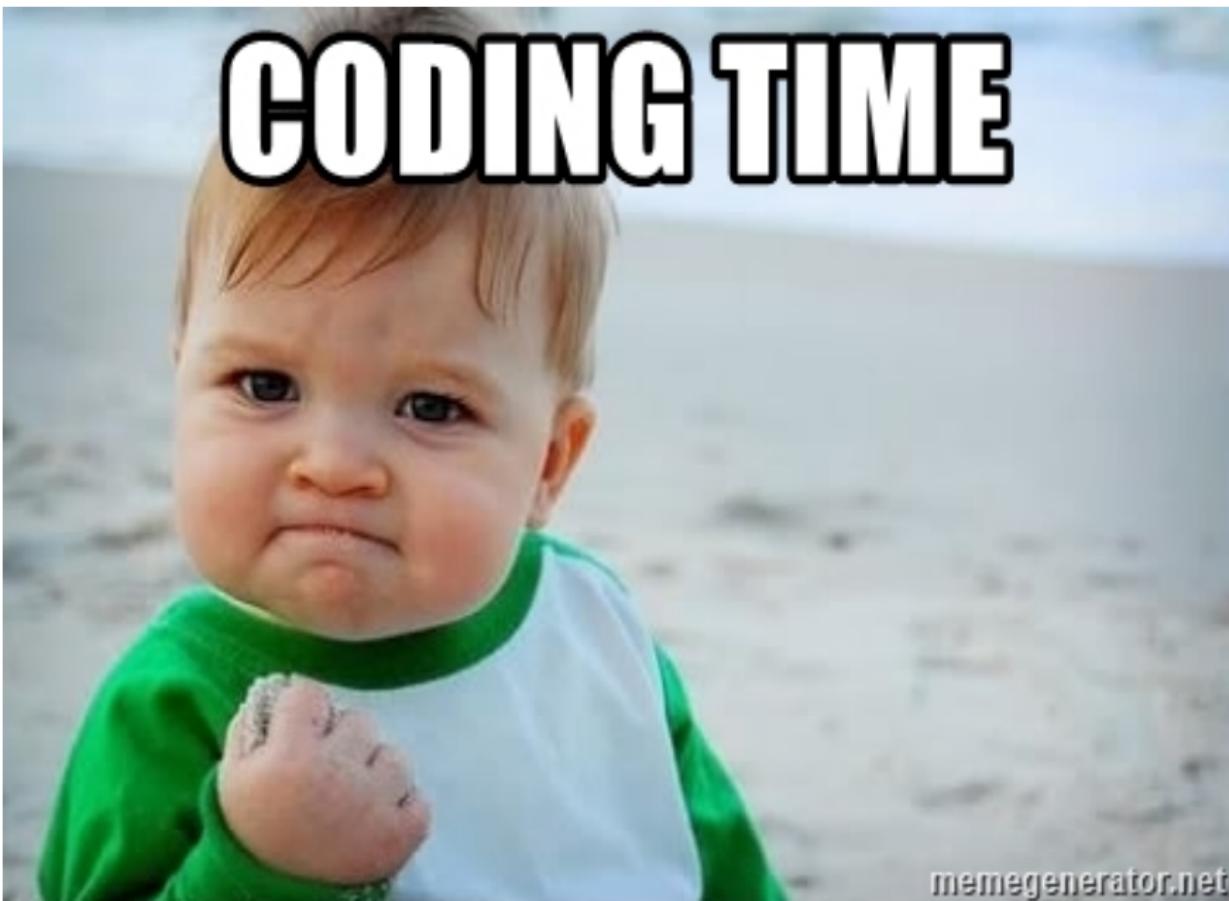
- Depends on the random sampling of new points, so it can be unstable
- Fit of linear model can be inaccurate
  - But we can check the r-squared score to know if that's the case
- Relatively slow for a single observation, in particular with images

# Types of Data

Lime supports many types of data:

- Tabular
- Recurrent Tabular
- Image
- Text

# CODING TIME



memegenerator.net

# SHAP values

- Similar to LIME, the focus here is on local methods designed to explain a prediction  $f(x)$  based on a single input  $x$ .
- SHAP values are a unified measure of feature importance.
- The main idea comes from the game theory, where one wants to know how much is the contribution of each individual to the cooperation.

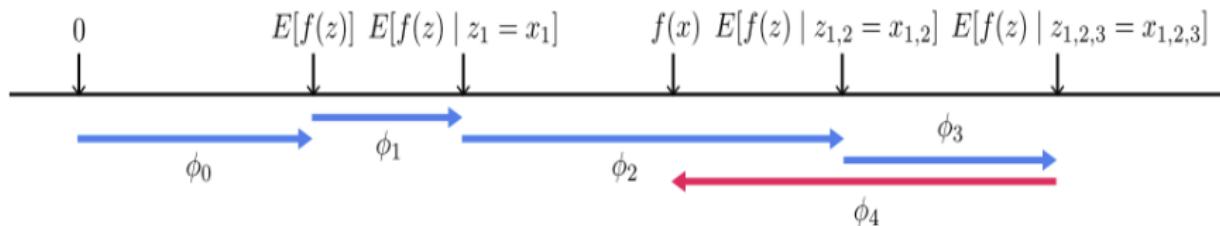
# SHAP values

- Here, for a specific vector  $x^*$ , we decompose the prediction  $f(x^*)$

$$f(x^*) = \phi_0 + \sum_{j=1}^p \phi_j$$

where  $\phi_0 = E[f(x)]$ ,  $\phi_j$  is the SHAP value, i.e., contribution of each feature.

- That is, the Shapley values explain the difference between the prediction  $f(x^*)$  and the global average prediction.



# SHAP values

- As we saw the contribution of each feature depends on the order and the subsets of features.
- Let's denote the contribution function of the subset  $S$  as

$$v(S) = E[f(x)|x_S = x_S^*]$$

- For example, when the number of features are  $\{1, 2, 3\}$  then there are 8 possible subsets  $(\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\})$ .
- The contribution of  $\{2, 3\}$  is  $v(\{2, 3\}) = E[f(x)|x_{\{2,3\}}^*]$

# SHAP values

- It can be shown that for a linear regression model  $f(x) = \beta_0 + \sum_{j=1}^p \beta_j x_j$ , where all features are independent, the SHAP values are

$$\phi_0 = \beta_0 + \sum_{j=1}^p \beta_j E[x_j], \text{ and } \phi_j = \beta_j (x_j^* - E[x_j]), \quad j = 1, \dots, p$$

- For general models, there is no such closed form and computation of all possible subsets is expensive.
- [6] that solution can be transformed to a weighted least square problem

$$\sum_{S \subset \{1, \dots, p\}} (v(S) - (\phi_0 + \sum_{j \in S} \phi_j))^2 k(p, S)$$

where the kernel

$$k(p, S) = \frac{(p-1)}{\binom{p}{|S|} |S|(p-|S|)}$$

# Summary

- Advantages
  - Solid theoretical foundation in game theory
  - The prediction is fairly distributed among the feature values.
  - SHAP connects LIME and Shapley values.
- Disadvantages
  - Computationally expensive
  - Ignores feature dependence.
  - It is possible to create intentionally misleading interpretations with SHAP [12].

# References I

-  J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim.  
Sanity checks for saliency maps.  
In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9525–9536, 2018.
-  S. Beery, G. Van Horn, and P. Perona.  
Recognition in terra incognita.  
In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, 2018.
-  P. Biecek and T. Burzykowski.  
*Explanatory Model Analysis*.  
Chapman and Hall/CRC, New York, 2021.
-  A. Chatopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian.  
Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks.  
*CoRR*, 2017.

# References II

-  A. Kapishnikov, T. Bolukbasi, F. B. Viégas, and M. Terry.  
Segment integrated gradients: Better attributions through regions.  
*CoRR*, abs/1906.02825, 2019.
-  S. M. Lundberg and S.-I. Lee.  
A unified approach to interpreting model predictions.  
NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.
-  C. Molnar.  
*Interpretable Machine Learning*.  
2 edition, 2022.
-  M. T. Ribeiro, S. Singh, and C. Guestrin.  
"why should i trust you?": Explaining the predictions of any classifier.  
In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1135–1144, 2016.

# References III

-  M. T. Ribeiro, S. Singh, and C. Guestrin.  
"why should i trust you?": Explaining the predictions of any classifier.  
In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1135–1144, 2016.
-  C. Rudin.  
*Nature Machine Intelligence*, 1(5):206–215, 2019.
-  A. Shrikumar, P. Greenside, and A. Kundaje.  
Learning important features through propagating activation differences.  
*CoRR*, abs/1704.02685, 2017.
-  D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju.  
How can we fool LIME and shap? adversarial attacks on post hoc explanation methods.  
*CoRR*, abs/1911.02508, 2019.

# References IV

-  D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg.  
Smoothgrad: removing noise by adding noise.  
*CoRR*, 2017.
-  J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller.  
Striving for simplicity: The all convolutional net, 2014.
-  M. Sundararajan, A. Taly, and Q. Yan.  
Axiomatic attribution for deep networks.  
*CoRR*, abs/1703.01365, 2017.
-  M. D. Zeiler and R. Fergus.  
Visualizing and understanding convolutional networks, 2013.
-  B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba.  
Learning deep features for discriminative localization, 2015.

# Example Details

- Logistic regression models a relationship between predictor variables and a categorical response variable.
- Logistic regression helps us estimate a probability of falling into a certain level of the categorical response given a set of predictors.
- The multiple binary logistic regression model is the following:

$$P(Y = 1) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k))}$$

- and its logit transformation

$$\log\left(\frac{P(Y = 1)}{P(Y = 0)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

[Back to Slides](#)