Explainability techniques for black-box decision rules in machine-learning

Laurent Risser
CNRS research engineer at *Institut de Mathématiques de Toulouse* and *3IA ANITI*

lrisser@math.univ-toulouse.fr

Machine Learning (M.L.):
- **Automatic predictions** based on **decision rules** defined during a training phase.
- **Training** consists in **tuning the parameters** of a predefined decision rules model so that it mimics at best the decisions made **in a training set.**

Artificial Intelligence (A.I.):
- Applications of M.L. and Logics (e.g. expert systems)

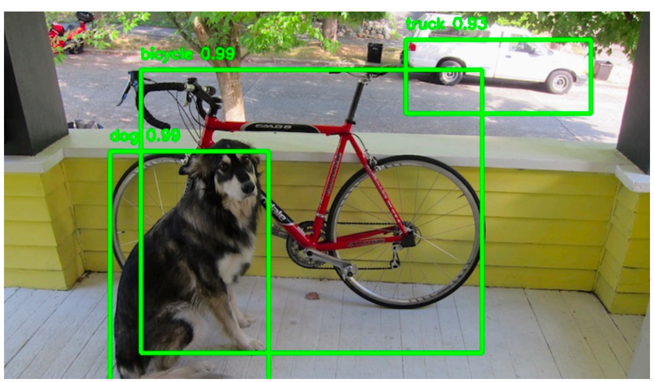
Models based on **simple decision rules**
- Linear Models
- Decision trees

Models based on decision rules that are **not very interpretable**
- Kernel SVM
- Random forests

Models **even less interpretable**
- Deep neural networks

Years

Beginning of the 1980s: Expert systems for aiding the piloting. Input data are 10s of sensors.



End of 2010s : Real time detection of more than 1000 image features using CNNs in 24 fps videos (Yolo v3).

- How does an explainable prediction model works?
- How does a Convolutional Neural Network works?
- Need for explainability for Black-Box prediction models
- Three explainability techniques in Machine Learning
    A. Lime
    B. Grad-CAM
    C. Entropic Variable Boosting

1) Introduction

Classic example: The MNIST database *[LeCun and Cortes, 2010]*

Examples of images X_i

Data:

- *60K* training images $\{X_i\}_{i=1,\ldots,60000}$ of *24x24* pixels
- Each image X_i represents a handwritten digit.
- A label $Y_i \in \{0,1,\ldots,9\}$ is associated to each X_i

Prediction model:

- $g_\theta : \mathbb{R}^{24*24} \mapsto [0,1]^{10}$
- Takes an image as input
- Returns a vector of size 10 representing the probability of being in each class

  (e.g $\bar{Y}_i = (0,0,1,0,\ldots,0)$ if $Y_i = 2$)

Training the parameters θ:

- We optimise:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{60000} \sum_{i=1}^{60000} || g_\theta(X_i) - \bar{Y}_i ||_2^2$$

Prediction on a new image: $\widehat{Y_{new}} = g_{\hat{\theta}}(X_{new})$

1) Introduction — Explainable prediction model

<u>Example of fully explainable model → linear model</u>:

- $g_{\hat{\theta}}(X_{new})[i]$ is the predicted probability that X_{new} represents the digit i.

- We denote $p \in \Omega$ the image pixels

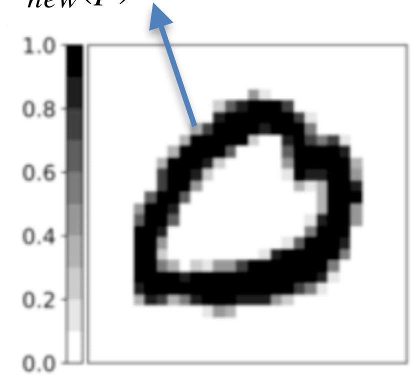
- Here, the parameters θ are a set of weights for each pixel:

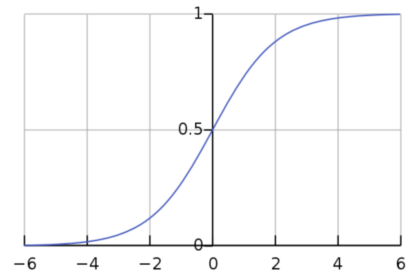
$$\Theta = \left\{ w_0(0{,}0), w_0(0{,}1), \ldots w_0(28{,}28), w_1(0{,}0), \ldots, w_9(28{,}28) \right\}$$
$$= \left\{ \{w_0(p)\}_{p \in \Omega}, \{w_1(p)\}_{p \in \Omega}, \ldots, \{w_9(p)\}_{p \in \Omega} \right\}$$

- Logistic regression model:

$$g_{\hat{\theta}}(X_{new})[i] = \varphi\left(\sum_{p \in \Omega} X_{new}(p) w_i(p) \right)$$

$X_{new}(p) = 0.23$





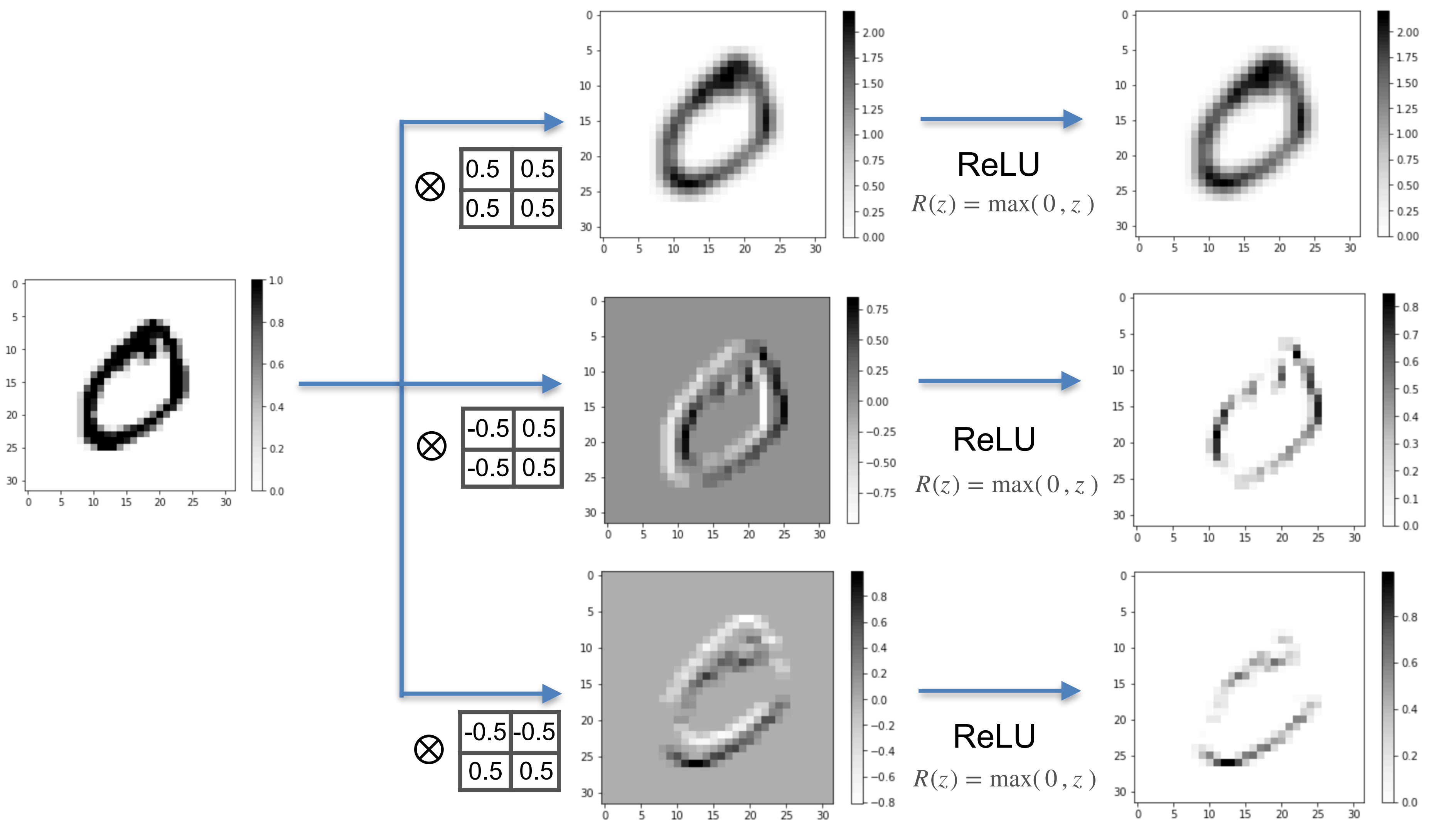
Logistic function φ

w_0 w_1 w_2 w_3 w_4 w_9

→About 91% accuracy on the test set of 10K images.

Example of clearly unexplainable model → convolutional neural network:

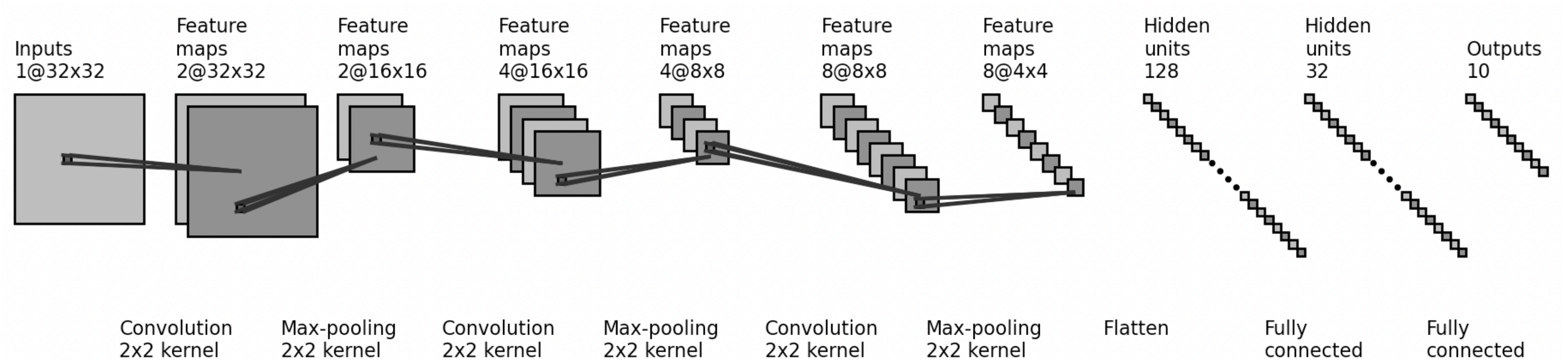
Convolutional Neural-Networks (CNNs) heavily use convolutional filters and the ReLU function, e.g.:



Example of clearly unexplainable model → convolutional neural network:

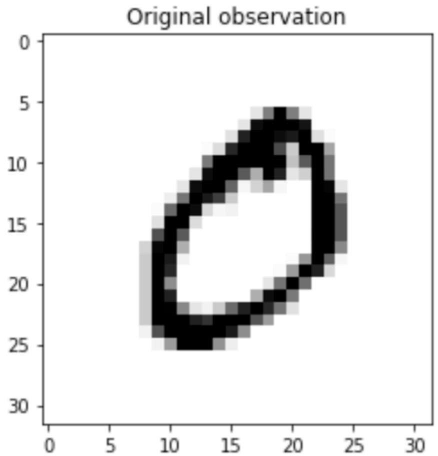
```
class basicCNN(nn.Module):
    def __init__(self):
        super(basicCNN, self).__init__()
        #Convolution/ReLU/MaxPooling layers
        self.conv1 = nn.Conv2d(1, 2, kernel_size=2, stride=1, padding=1) #1 to 2 channels
        self.pool1 = nn.MaxPool2d(kernel_size=2, stride=2)  #32x32 to 16x16
        self.conv2 = nn.Conv2d(2, 4, kernel_size=2, stride=1, padding=1) #2 to 4 channels
        self.pool2 = nn.MaxPool2d(kernel_size=2, stride=2)  #16x16 to 8x8
        self.conv3 = nn.Conv2d(4, 8, kernel_size=2, stride=1, padding=1) #4 to 8 channels
        self.pool3 = nn.MaxPool2d(kernel_size=2, stride=2)  #8x8 to 4x4

        #Dense layers
        self.fc1 = nn.Linear(8 * 4 * 4, 32)
        self.fc2 = nn.Linear(32, 10)
    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = self.pool1(x)
        x = F.relu(self.conv2(x))
        x = self.pool2(x)
        x = F.relu(self.conv3(x))
        x = self.pool3(x)
        x = x.view(-1, 8*4*4) #flatten the data
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return(x)
```



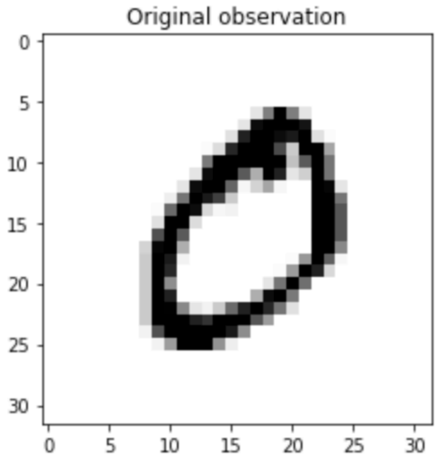
https://github.com/gwding/draw_convnet

Example of clearly unexplainable model → convolutional neural network:



X_i

Example of clearly unexplainable model → convolutional neural network:



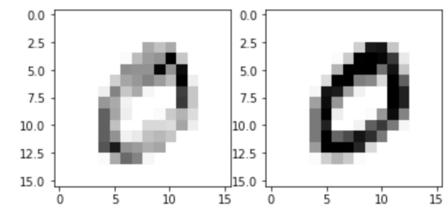
X_i

Convolution with

```
Filter 1:
 [[ 0.17653309  0.08818832]
 [-0.2810376   0.22499529]]
Bias 1:
 0.029923715
```

```
Filter 2:
 [[0.15315333 0.14285438]
 [0.22419561 0.16171551]]
Bias 2:
 -0.085632496
```

then ReLU, then MaxPooling (non-linear subsampling)



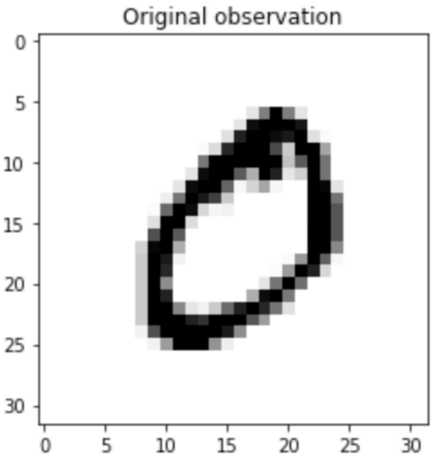
Convolution and ReLU:

Layer | output channel | Bias term for output channel c of layer 2 | Filter for output channel c of layer 2

$$I^{2,c}(p_x, p_y) = b_c^2 + \sum_{k=0}^{1} \sum_{l=0}^{1} X_i(p_x + k, p_y + l) w_c^2(k, l)$$

$$I^{2,c}(p_x, p_y) = \max(0\,, I^{2,c}(p_x, p_y))$$

Example of clearly unexplainable model → convolutional neural network:



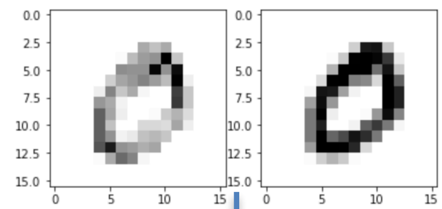
X_i

Convolution with

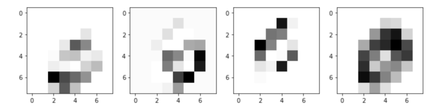
```
Filter 1:
 [[ 0.17653309  0.08818832]
 [-0.2810376   0.22499529]]
Bias 1:
 0.029923715
```

```
Filter 2:
 [[0.15315333 0.14285438]
 [0.22419561 0.16171551]]
Bias 2:
 -0.085632496
```

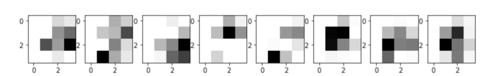
then ReLU, then MaxPooling (non-linear subsampling)



Convolution with 2×4 convolution filters
then ReLU, then Max-pooling



Convolution with 4×8 convolution filters
then ReLU, then Max-pooling



So-called Rectified Convolution Feature Maps

Convolution and ReLU:

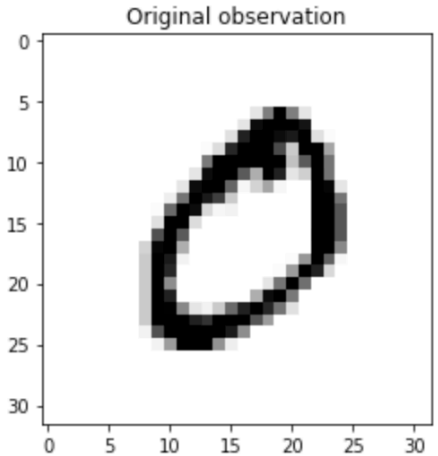
Sum over the channels of the input layer

Filter from channel c' of layer 2 to channel c of layer 3

$$I^{3,c}(p_x, p_y) = b_c^3 + \sum_{c'=1}^{2} \left(\sum_{k=0}^{1} \sum_{l=0}^{1} I^{2,c'}(p_x + k, p_y + l) w_{c,c'}^3(k, l) \right)$$

$$I^{3,c}(p_x, p_y) = \max(0\, , I^{3,c}(p_x, p_y))$$

Example of clearly unexplainable model → convolutional neural network:



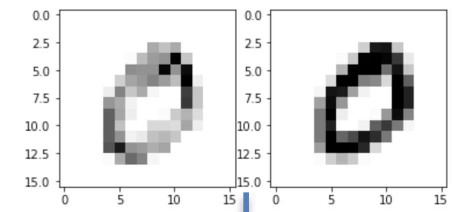
X_i

Convolution with

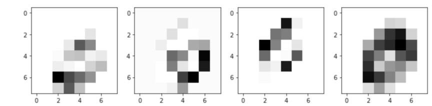
```
Filter 1:
 [[ 0.17653309  0.08818832]
 [-0.2810376   0.22499529]]
Bias 1:
 0.029923715
```

```
Filter 2:
 [[0.15315333 0.14285438]
 [0.22419561 0.16171551]]
Bias 2:
 -0.085632496
```

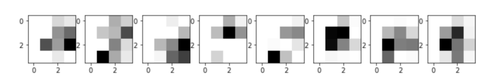
then ReLU, then MaxPooling (non-linear subsampling)



Convolution with 2×4 convolution filters
then ReLU, then Max-pooling



Convolution with 4×8 convolution filters
then ReLU, then Max-pooling

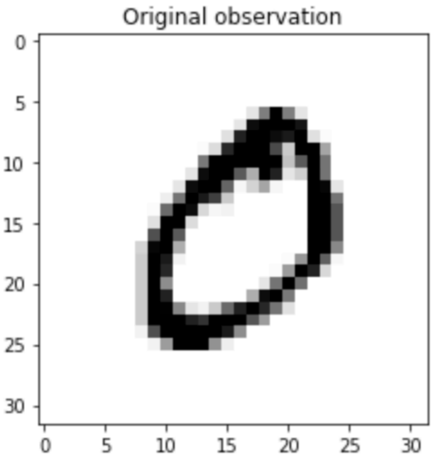


Flatten the 8 images of 4×4 pixels

Vector of size $8 \times 4 \times 4 = 128$

So-called *feature space* or *lattent space*

<u>Example of clearly unexplainable model → convolutional neural network</u>:



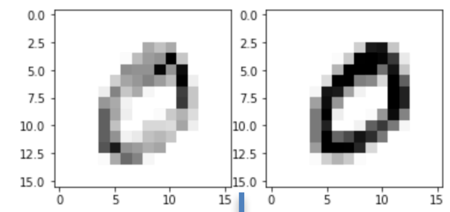
X_i

Convolution with

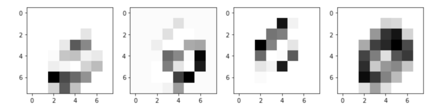
```
Filter 1:
 [[ 0.17653309  0.08818832]
 [-0.2810376   0.22499529]]
Bias 1:
 0.029923715
```

```
Filter 2:
 [[0.15315333 0.14285438]
 [0.22419561 0.16171551]]
Bias 2:
 -0.085632496
```

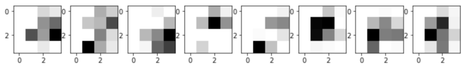
then ReLU, then MaxPooling (non-linear subsampling)



Convolution with 2×4 convolution filters
then ReLU, then Max-pooling



Convolution with 4×8 convolution filters
then ReLU, then Max-pooling



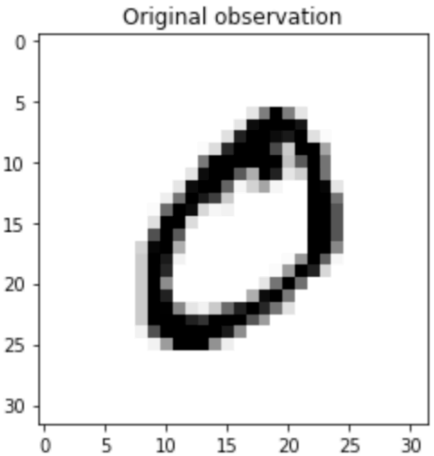
Flatten the 8 images of 4×4 pixels

Vector of size $8 \times 4 \times 4 = 128$

Product with a matrix of size 32×128 then ReLU

Vector of size 32

Example of clearly unexplainable model → convolutional neural network:



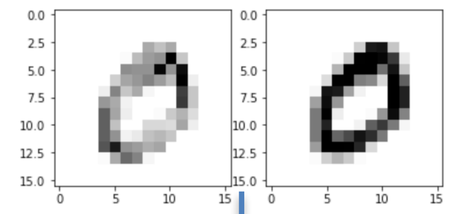
X_i

Convolution with

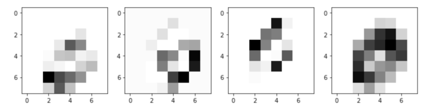
```
Filter 1:
 [[ 0.17653309  0.08818832]
  [-0.2810376   0.22499529]]
Bias 1:
 0.029923715
```

```
Filter 2:
 [[0.15315333 0.14285438]
  [0.22419561 0.16171551]]
Bias 2:
 -0.085632496
```

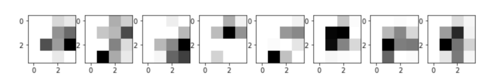
then ReLU, then MaxPooling (non-linear subsampling)



Convolution with 2×4 convolution filters
then ReLU, then Max-pooling



Convolution with 4×8 convolution filters
then ReLU, then Max-pooling



Flatten the 8 images of 4×4 pixels

Vector of size $8 \times 4 \times 4 = 128$

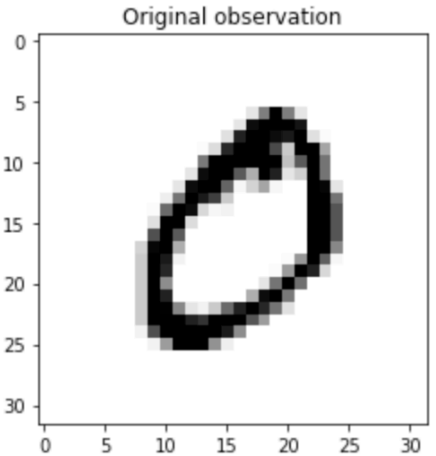
Product with a matrix of size 32×128 then ReLU

Vector of size 32

Product with a
matrix of size 10×32

$$\widehat{Y_i} = g_\theta(X_i) = (0.97, 0.02, \ldots, 0.07)$$

<u>Example of clearly unexplainable model → convolutional neural network</u>:



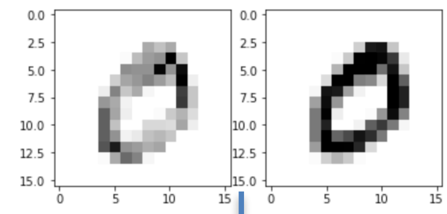
X_i

Convolution with

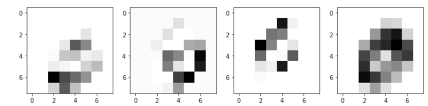
```
Filter 1:
 [[ 0.17653309  0.08818832]
 [-0.2810376   0.22499529]]
Bias 1:
 0.029923715
```

```
Filter 2:
 [[0.15315333 0.14285438]
 [0.22419561 0.16171551]]
Bias 2:
 -0.085632496
```

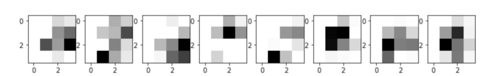
then ReLU, then MaxPooling (non-linear subsampling)



Convolution with 2×4 convolution filters
then ReLU, then Max-pooling



Convolution with 4×8 convolution filters
then ReLU, then Max-pooling



Flatten the 8 images of 4×4 pixels

Vector of size $8 \times 4 \times 4 = 128$

Product with a matrix of size 32×128 then ReLU

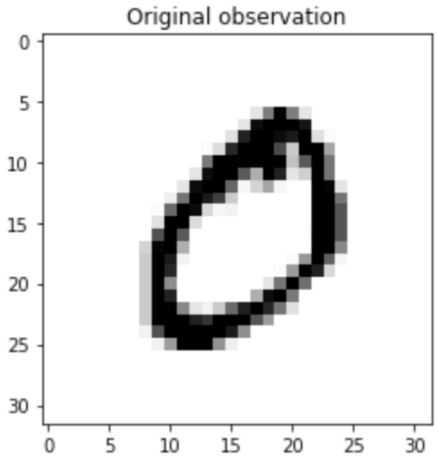
Vector of size 32

Product with a
matrix of size 10×32

$$\widehat{Y_i} = g_\theta(X_i) = (0.97, 0.02, \ldots, 0.07)$$

→About 96% accuracy here on the test set of 10K images. Can be improved to ≈99% accuracy with CNNs.

Example of clearly unexplainable model → convolutional neural network:



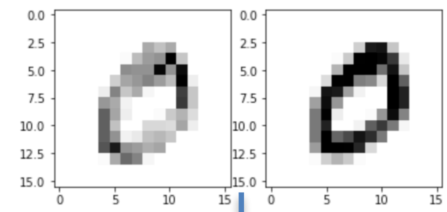
X_i

Convolution with

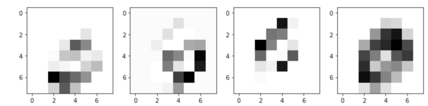
```
Filter 1:
 [[ 0.17653309  0.08818832]
 [-0.2810376   0.22499529]]
Bias 1:
 0.029923715
```

```
Filter 2:
 [[0.15315333 0.14285438]
 [0.22419561 0.16171551]]
Bias 2:
 -0.085632496
```

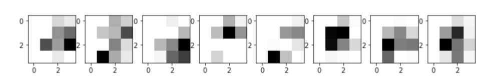
then ReLU, then MaxPooling (non-linear subsampling)



Convolution with 2×4 convolution filters
then ReLU, then Max-pooling



Convolution with 4×8 convolution filters
then ReLU, then Max-pooling



Flatten the 8 images of 4×4 pixels

Vector of size $8 \times 4 \times 4 = 128$

Product with a matrix of size 32×128 then ReLU

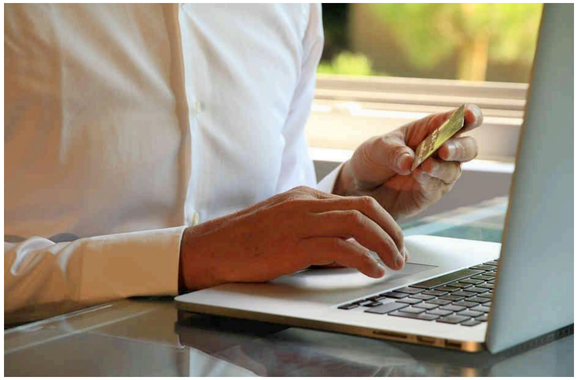
Vector of size 32

Product with a matrix of size 10×32

$$\widehat{Y_i} = g_{\boxed{\theta}}(X_i) = (0.97, 0.02, \ldots, 0.07)$$

Parameters θ → how to explain their influence or more generally why a decision was taken???

Explainability is not a big deal for many applications that made the use of Neural Networks popular *e.g.* for advertising or search of visual contents on the internet… BUT NNs are now used for many applications

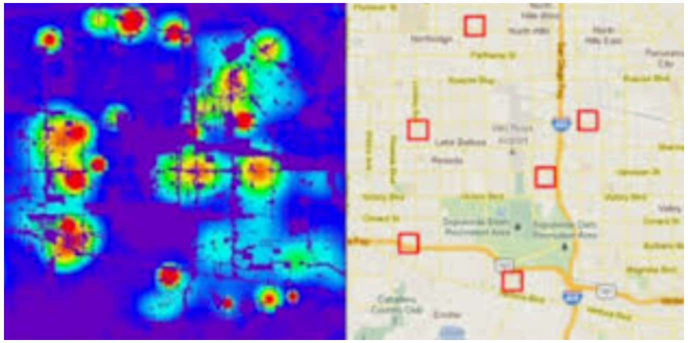


Online advertising

Information flows

Diagnostic

Autonomous vehicles



Predictive policing

…

Emergence of a *Right to explanation*

- E.U. (RGPD, art 22 — 2018) : « Right not to be subject to a decision solely based on automated processing, including profiling »
- Fr (Loi Informatique et Libertés) : « Right to understand the rules of automatic treatments and their main characteristics »
- NYC Bill (Dec. 2017) : Local laws related to automatic decision systems

Exemples of recent works

- Edwards, Veal : *Enslaving the Algorithm : From a « Right to an Explanation » to a « Right to Better Decisions »* IEEE Security and Privacy 16(3), 2018
- Besse, Castet-Renard, Garivier, Loubes : L'I.A. du quotidien peut-elle être éthique? Statistique et société 6(3), 2018 — https://www.youtube.com/watch?v=RwsMv0ILxos
- Castet-Renard, Besse, Loubes, Perussel : Encadrement des risques techniques et juridiques des activités de police prédictive. Rapport CHEMI du Ministère de l'Intérieur, 2019
- ACM-FAT* community
- …

Strong interest in industry as well → robust decision making + towards certifiable IA

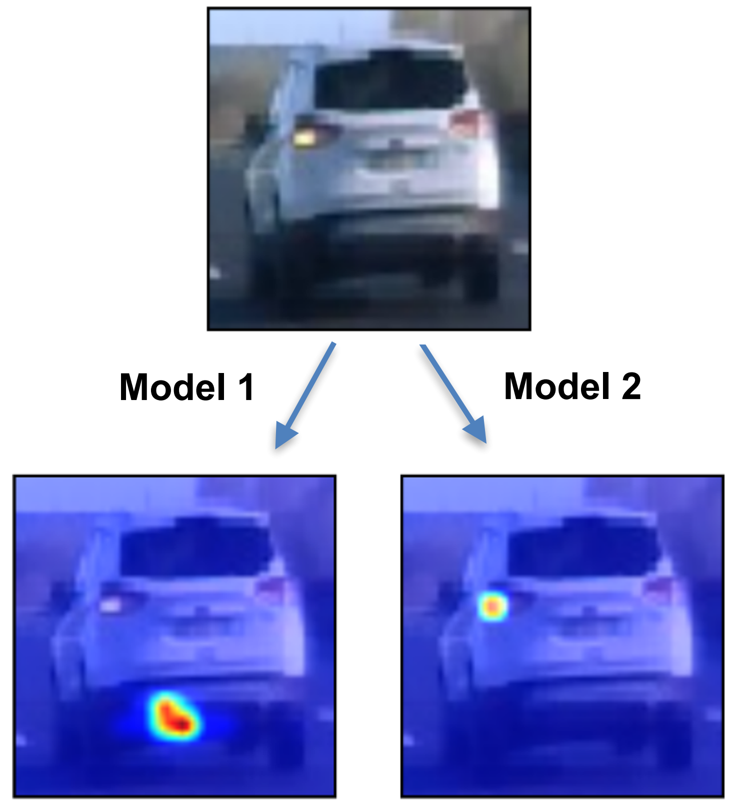
No blink
(But possibly break lights)

Left blink

Right blink

Warning

Example:



Suppose that the predictions are generally accurate:
- Which features were used to take the decision?
- If inadequate features were used, the NN is likely to generalise poorly!

2) Need for explainability

"Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI", A. Barrieta et al, 2019

"Interpretable Explanations of Black Boxes by Meaningful Perturbation", Ruth C. Fong, Andrea Vedaldi, 2017

"*MAGIX: model agnostic globally interpretable explanations*," N. Puri, P. Gupta, P. Agarwal, S. Verma, and B. Krishnamurthy, CoRR,vol. abs/1706.07160, 2017.

"Why should I trust you? Explaining the predictions of any classifier.", T. Ribeiro, S. Singh, and C. Guestrin, 2016 - International Conference on Knowledge Discovery and Data Mining, ACM2016

"Local Rule-Based Explanations of Black Box Decision Systems" (LORE),Riccardo Guidotti et al 2018,

"Anchors: High-precision model-agnostic explanations," T. Ribeiro, S. Singh, and C. Guestrin, , in AAAI Conference on Artificial Intelligence, 2018.

"Visualizing the feature importance for black box models", G. Casalicchio, C. Molnar, B. Bischl, arXiv:1804.06620.

"*Auditing black-box models for indirect influence"*, P. Adler, C. Falk, S. A. Friedler, T. Nix, G. Rybeck, C. Scheidegger, B. Smith, S. Venkatasubramanian, Knowledge and Information Systems 54 (1) (2018) 95–122.

"*Entropic Variable Projection for Explainability and Intepretability"*, F. Bachoc and F. Gamboa and M. Halford and J.-M. Loubes and L. Risser, 2018, arXiv:1810.07924.

"Grad-cam: Visual explanations from deep networks via gradient-based localization", R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

"Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning" , N. Papernot, P. McDaniel, (2018). arXiv:1803.04765.

"Interpretable convolutional neural networks", Q. Zhang, Y. Nian Wu, S.-C. Zhu, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8827–8836.

"InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets" , X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, (2016). arXiv:1606.03657

"Not just a black box: Learning important features through propagating activation differences", Shrikumar, P. Greenside, A. Shcherbina, A. Kundaje, 2016, arXiv:1605.01713

"Interpretable explanations of black boxes by meaningful perturbation", R. C. Fong, A. Vedaldi, in: IEEE International Conference on Computer Vision, 2017, pp. 3429–3437.

"On the Robustness of Interpretability Methods", Alvarez-Melis et T. S. Jaakkola, *arXiv:1806.08049 [cs, stat]*, juin 2018.

"Interpretable Deep Learning under Fire", X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, et T. Wang, *arXiv:1812.00891 [cs]*, sept. 2019.

"Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients", S. Ross et F. Doshi-Velez, arXiv:1711.09404 [cs], nov. 2017.

"Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR", Wachter, B. Mittelstadt, et C. Russell, SSRN Journal, 2017.

… and many others …

"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

https://arxiv.org/pdf/1602.04938.pdf
https://homes.cs.washington.edu/~marcotcr/blog/lime/
https://github.com/marcotcr/lime

LIME explains why a specific (local) prediction is made by using an explainable surrogate model

X_i^1 →
X_i^2 →
X_i^3 →
$\vdots$ →
X_i^5 →

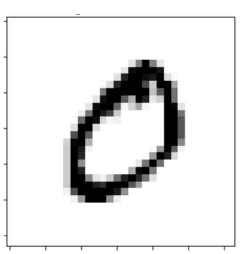
→ $\hat{Y}_i$

Input variables X_i — e.g.

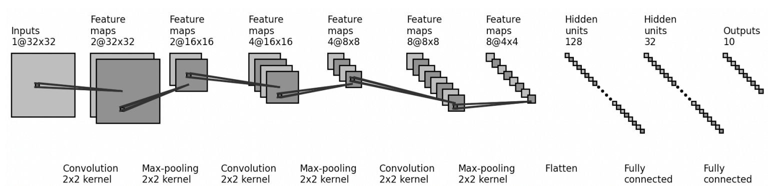
Trained black-box model f_θ — e.g.

Prediction $\hat{Y}_i = f_\theta(X_i)$ — e.g.

Intensities of



N.N. model



Output probabilities

$(0.98, 0.03, 0.1, \ldots, 0.08)^\intercal$

3) Three explainability solutions → LIME (Local interpretable model-agnostic explanations)

“Why Should I Trust You?”
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

https://arxiv.org/pdf/1602.04938.pdf
https://homes.cs.washington.edu/~marcotcr/blog/lime/
https://github.com/marcotcr/lime

LIME explains why a specific (local) prediction is made by using an explainable surrogate model

X_i^1 →
X_i^2 →
X_i^3 →
$\vdots$ →
X_i^5 →

→ $\hat{Y}_i$

Input variables X_i | **<u>Trained</u> black-box model** f_θ | **Prediction** $\hat{Y}_i = f_\theta(X_i)$

<u>Training a local surrogate models to explain the prediction of X_i with f_θ</u>:

- Randomly perturb $X_i \rightarrow \{X_i^p\}_{p=1,\ldots,P}$
- Define a distance for the perturbed observations $\pi_{X_i}(X_i^p) = dist(X_i, X_i^p)$.
- Consider an explainable model $g_{\theta'}$ (e.g. a linear model, a decision tree, …)
- Optimise the parameters θ' by minimising: $\sum_{p=1}^{P} \pi_{X_i}(X_i^p)(g_{\theta'}(X_i^p) - f_\theta(X_i^p))^2$
- Explain the prediction thanks to $g(\theta')$

3) Three explainability solutions → LIME (Local interpretable model-agnostic explanations)

"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

https://arxiv.org/pdf/1602.04938.pdf
https://homes.cs.washington.edu/~marcotcr/blog/lime/
https://github.com/marcotcr/lime

LIME explains why a specific (local) prediction is made by using an explainable surrogate model

X_i^1 →
X_i^2 →
X_i^3 →
$\vdots$ →
X_i^5 →

→ $\hat{Y}_i$

Input variables X_i | **Trained black-box model f_θ** | **Prediction $\hat{Y}_i = f_\theta(X_i)$**

Training a local surrogate models to explain the prediction of X_i with f_θ:

- Randomly perturb $X_i \rightarrow \{X_i^p\}_{p=1,\dots,P}$
- Define a distance for the perturbed observations $\pi_{X_i}(X_i^p) = dist(X_i, X_i^p)$.
- Consider an explainable model $g_{\theta'}$ (e.g. a linear model, a decision tree, …)
- Optimise the parameters θ' by minimising: $\sum_{p=1}^{P} \pi_{X_i}(X_i^p)(g_{\theta'}(X_i^p) - f_\theta(X_i^p))^2$
- Explain the prediction thanks to $g(\theta')$

In the image case, the pixel intensities are not necessarily independently perturbed!

Interpretable Components

"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

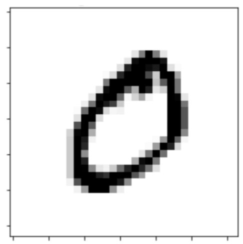
Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

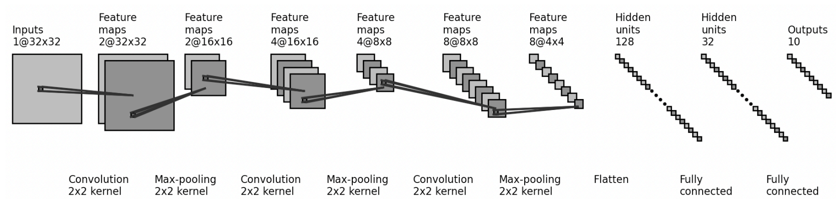
Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

https://arxiv.org/pdf/1602.04938.pdf
https://homes.cs.washington.edu/~marcotcr/blog/lime/
https://github.com/marcotcr/lime

To go back to our example:

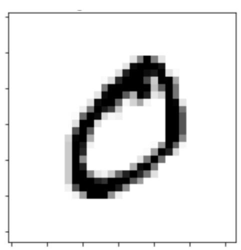
Our neural-network prediction model f_θ …

X_i 

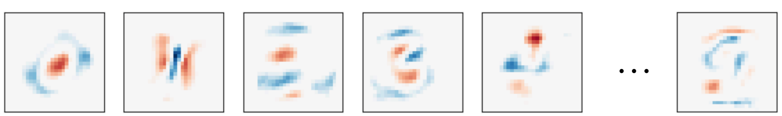


$f_\theta(X_i) = (0.97, 0.02, \ldots, 0.07)$

… can become a linear, and straightforwardly interpretable, model $g_{\theta'}$ for images close to X_i:

X_i 

Weighted sum of the intensities with weights:



(followed by logistic function)

$g_{\theta'}(X_i) = (0.95, 0.03, \ldots, 0.05)$

"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

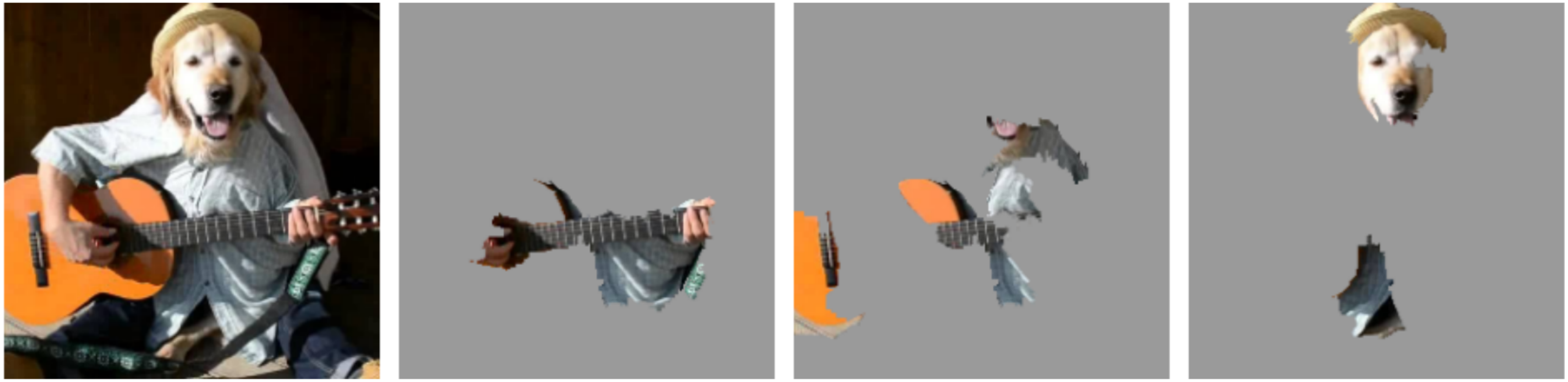
Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

https://arxiv.org/pdf/1602.04938.pdf
https://homes.cs.washington.edu/~marcotcr/blog/lime/
https://github.com/marcotcr/lime

Classic results out of the original LIME paper:



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

https://arxiv.org/pdf/1602.04938.pdf
https://homes.cs.washington.edu/~marcotcr/blog/lime/
https://github.com/marcotcr/lime

Classic results out of the original LIME paper:

(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

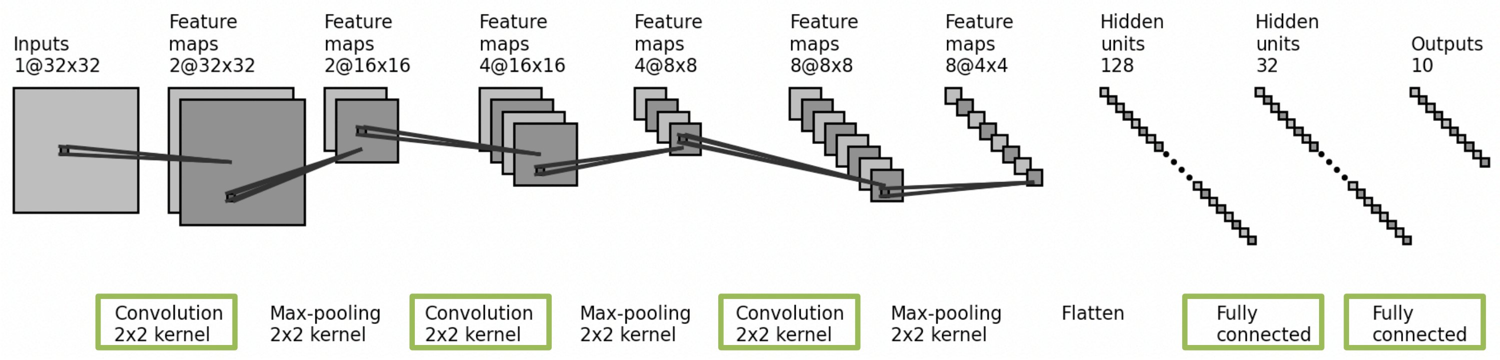
Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

To understand Grad-CAM, one must first have in mind how a N.N. is trained

- Training observations: $\{(X_i, Y_i)\}_{i=1,\dots,n}$
- $\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} loss(f_\theta(X_i), Y_i) = \arg\min_{\theta} R\left(f_\theta, \{(X_i, Y_i)\}_{i=1,\dots,n}\right)$
- Gradient descent based optimisation: $\theta_{it+1} = \theta_{it} - \lambda \nabla_\theta R\left(f_{\theta_{it}}, \{(X_i, Y_i)\}_{i=1,\dots,n}\right)$



Parameters θ

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

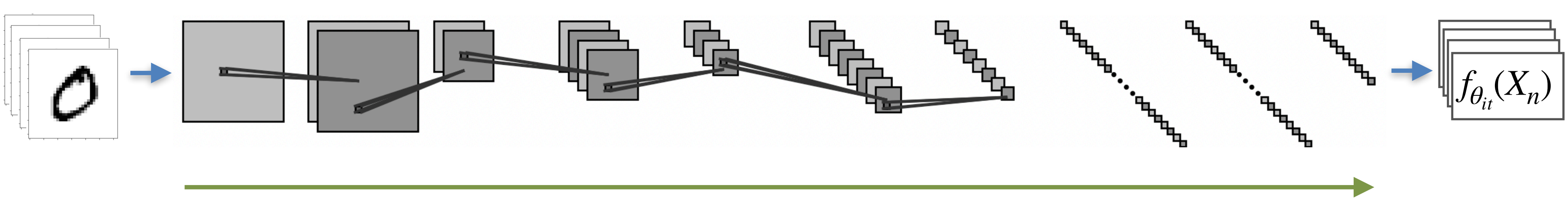
Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

To understand Grad-CAM, one must first have in mind how a N.N. is trained

- Training observations: $\{(X_i, Y_i)\}_{i=1,\ldots,n}$
- $\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} loss(f_\theta(X_i), Y_i) = \arg\min_{\theta} R\left(f_\theta, \{(X_i, Y_i)\}_{i=1,\ldots,n}\right)$
- Gradient descent based optimisation: $\theta_{it+1} = \theta_{it} - \lambda \nabla_\theta R\left(f_{\theta_{it}}, \{(X_i, Y_i)\}_{i=1,\ldots,n}\right)$

Gradient estimation



Make several predictions (mini-batch)

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

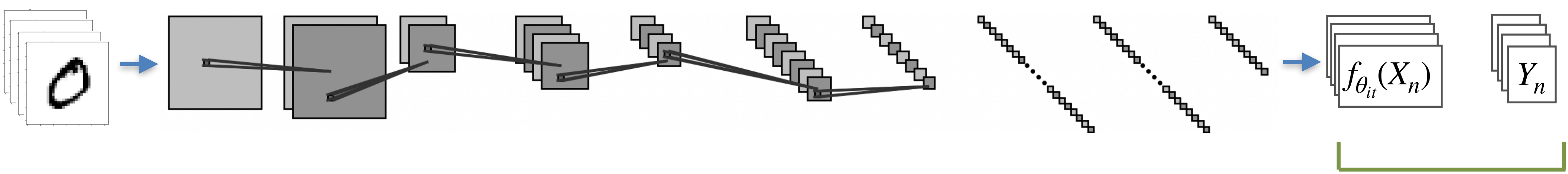
Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

To understand Grad-CAM, one must first have in mind how a N.N. is trained

- Training observations: $\{(X_i, Y_i)\}_{i=1,\ldots,n}$
- $\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} loss(f_\theta(X_i), Y_i) = \arg\min_{\theta} R\left(f_\theta, \{(X_i, Y_i)\}_{i=1,\ldots,n}\right)$
- Gradient descent based optimisation: $\theta_{it+1} = \theta_{it} - \lambda \nabla_\theta R\left(f_{\theta_{it}}, \{(X_i, Y_i)\}_{i=1,\ldots,n}\right)$

Gradient estimation



Approx. $R(\ldots)$ and its derivatives w.r.t. the N.N outputs

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

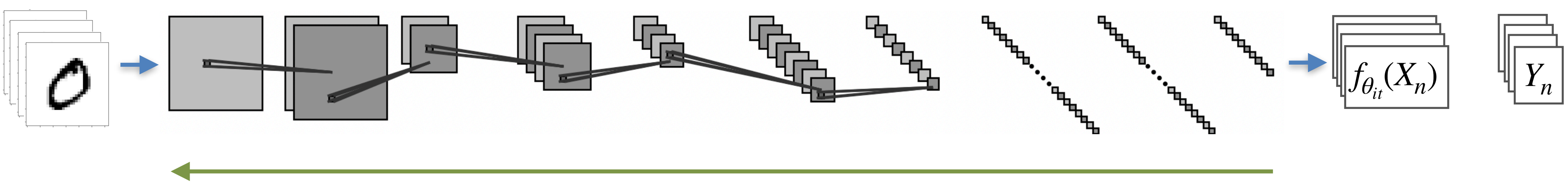
Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

To understand Grad-CAM, one must first have in mind how a N.N. is trained

- Training observations: $\{(X_i, Y_i)\}_{i=1,\ldots,n}$
- $\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} loss(f_\theta(X_i), Y_i) = \arg\min_{\theta} R\left(f_\theta, \{(X_i, Y_i)\}_{i=1,\ldots,n}\right)$
- Gradient descent based optimisation: $\theta_{it+1} = \theta_{it} - \lambda \nabla_\theta R\left(f_{\theta_{it}}, \{(X_i, Y_i)\}_{i=1,\ldots,n}\right)$

Gradient estimation



Back-propagate this information to compute the derivative of R w.r.t. all N.N. parameters

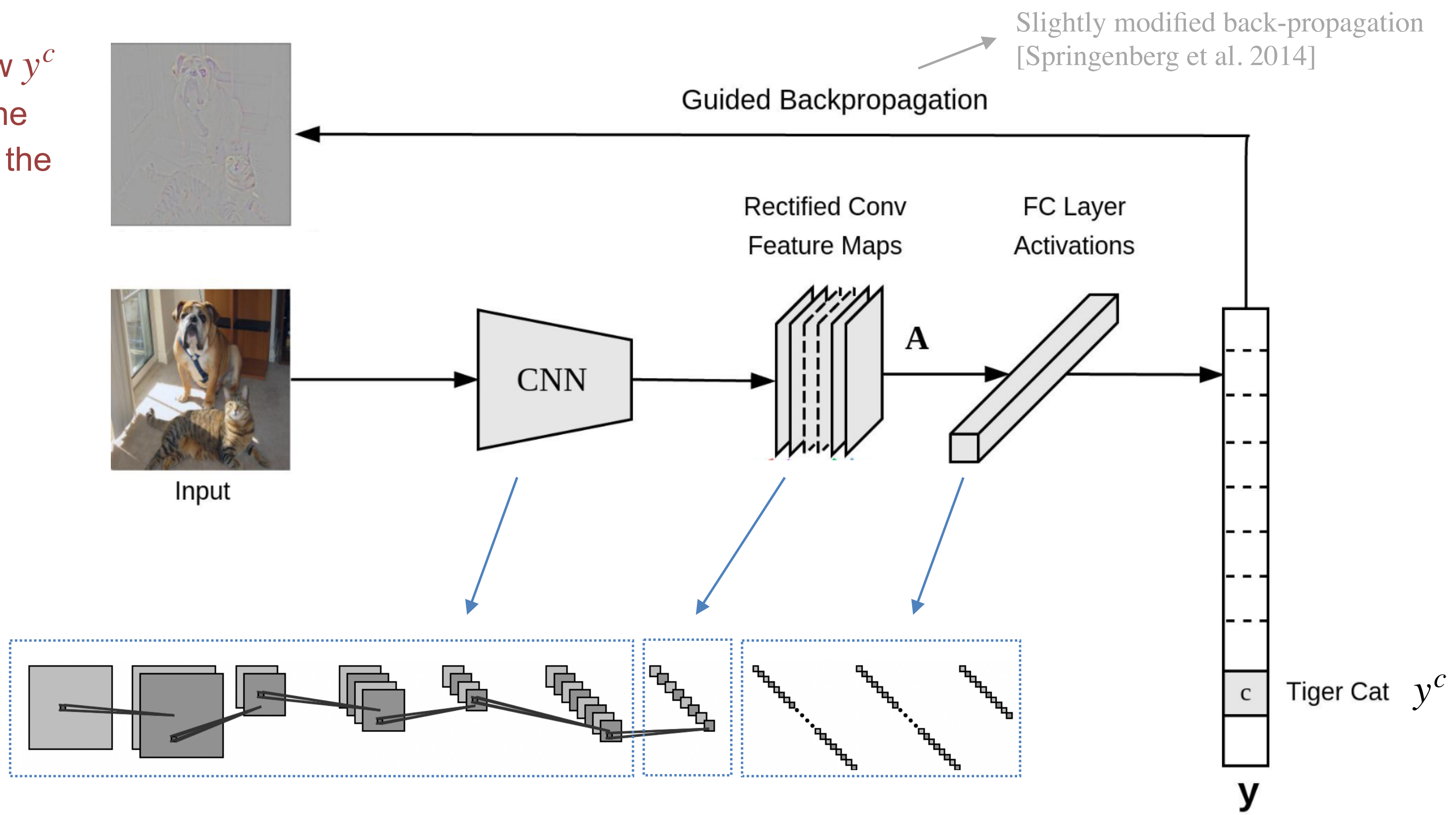
Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

Instead of back-propagating the derivatives of the risk R, it is possible to back-propagate the derivatives of a specific value in the N.N. outputs

Represents how y^c is sensitive to the N.N. inputs (for the tested image)



3) Three explainability solutions → Grad-CAM

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

Instead of back-propagating the derivatives of the risk R, it is possible to back-propagate the derivatives of a specific value in the N.N. outputs

GB for "Cat"

GB for "Dog"

Not that convincing … but a good starting point! → Not class-discriminative but high resolution

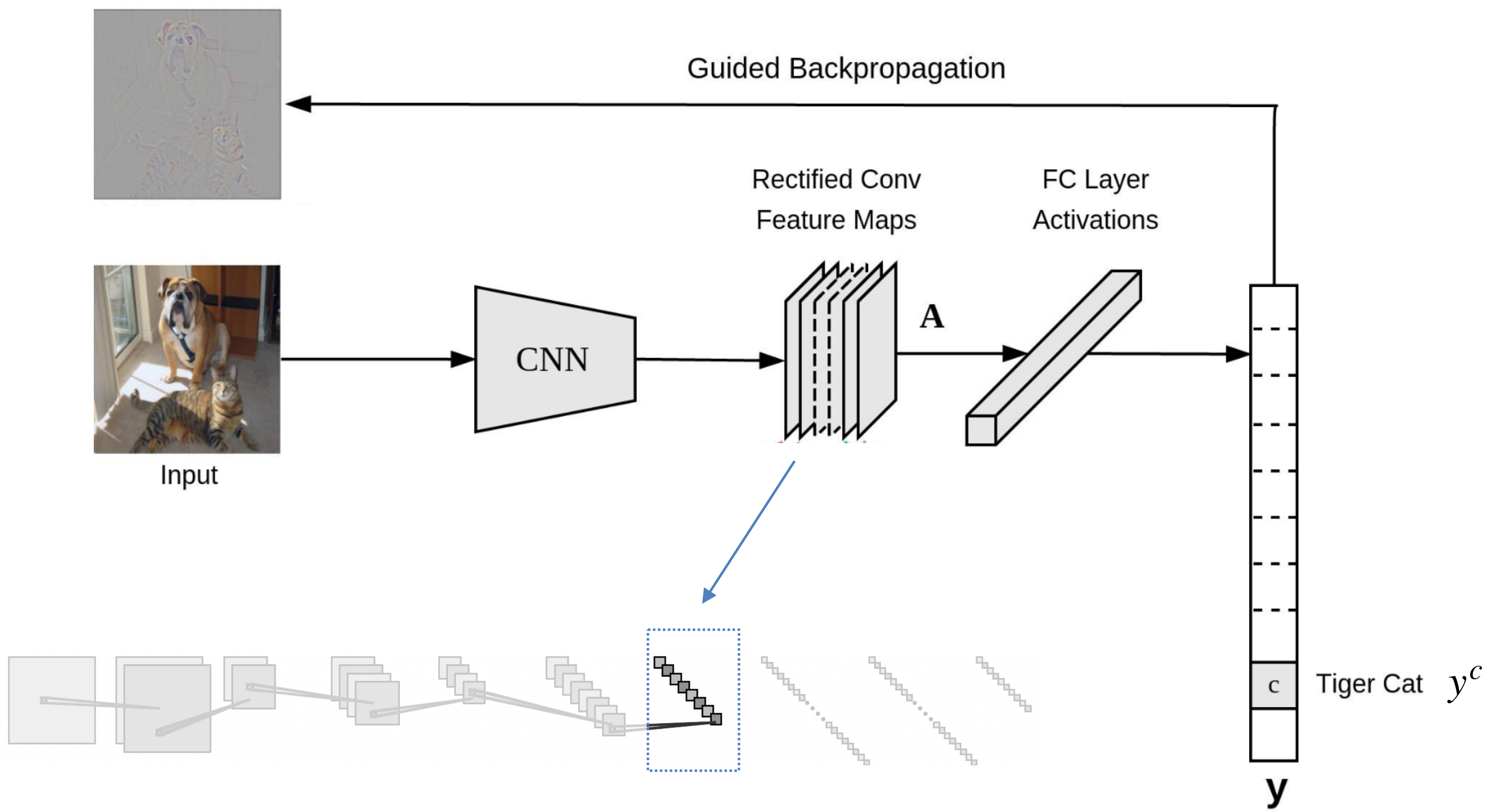
Grad-CAM will compute a special mask for this result

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

To get a more class discriminative Grad-CAM uses the Rectified Convolution Feature Maps $A^k_{i,j}$
(where k is a channel associated to a feature and (i,j) are coordinates in these subsampled images of detected features)



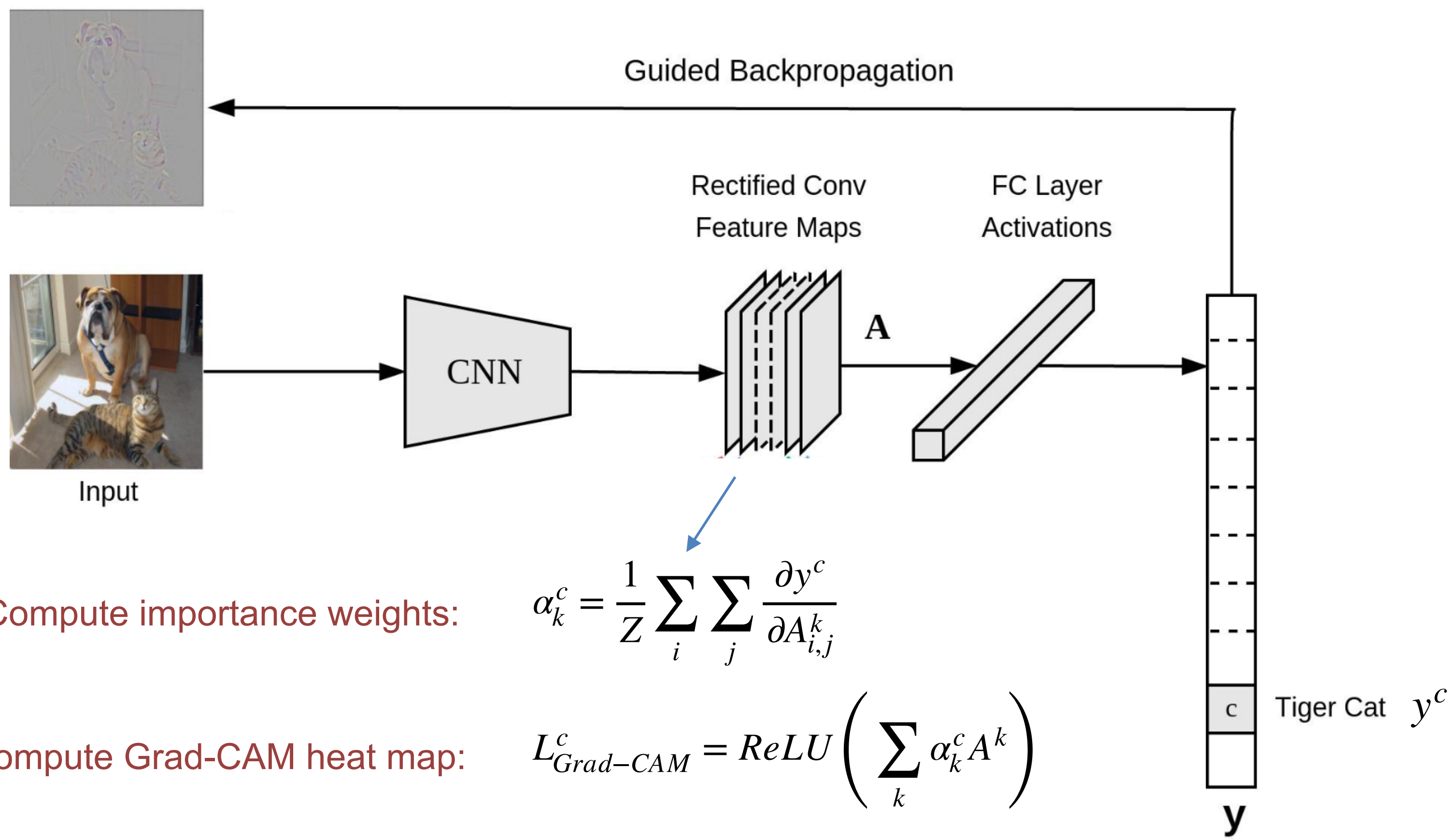
Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

To get a more class discriminative Grad-CAM uses the Rectified Convolution Feature Maps $A^k_{i,j}$

(where k is a channel associated to a feature and (i, j) are coordinates in these subsampled images of detected features)



Compute importance weights: $$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{i,j}}$$

Compute Grad-CAM heat map: $$L^c_{Grad-CAM} = ReLU\left(\sum_k \alpha_k^c A^k\right)$$

y^c

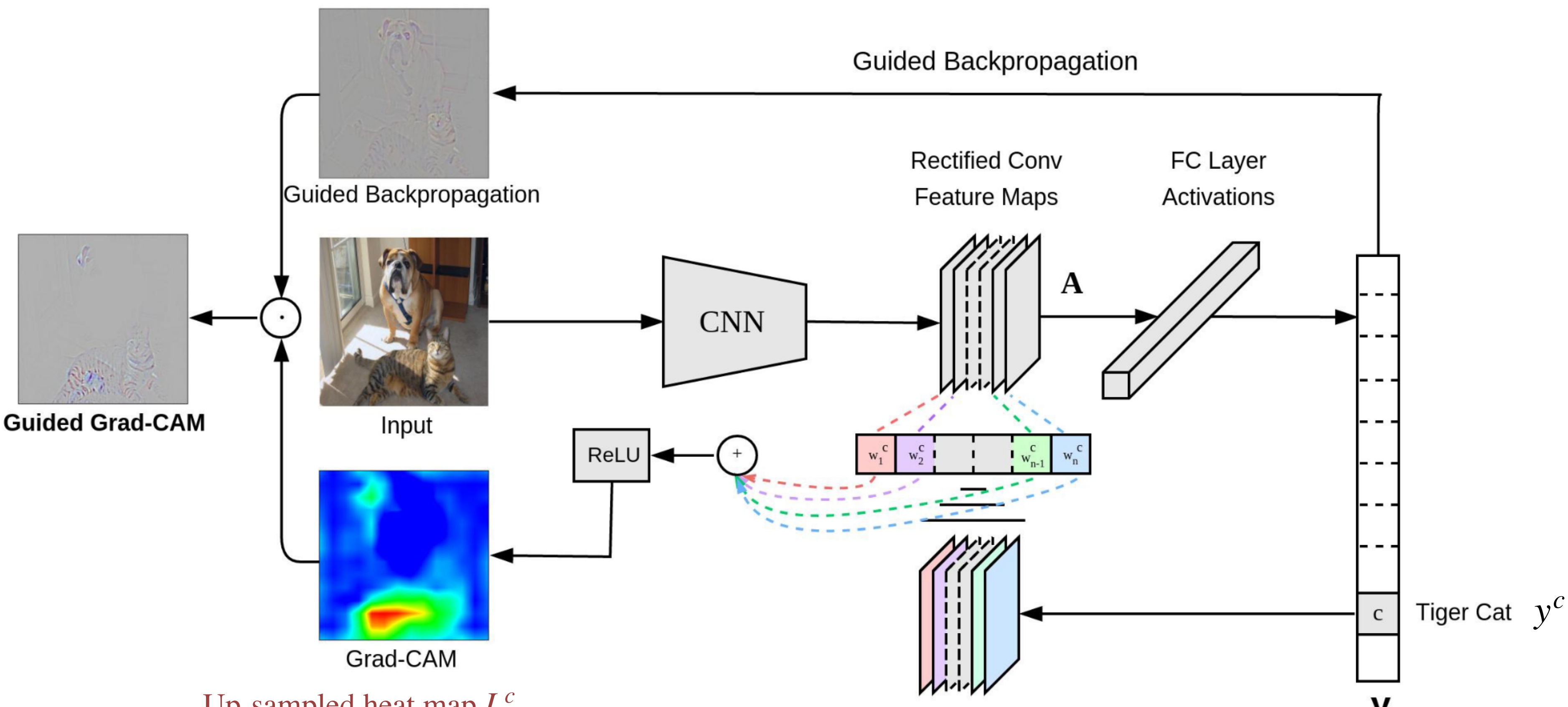
Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

To get a more class discriminative Grad-CAM uses the Rectified Convolution Feature Maps $A^k_{i,j}$

(where k is a channel associated to a feature and (i,j) are coordinates in these subsampled images of detected features)



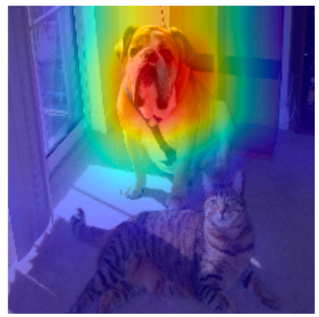
Up-sampled heat map $L^c_{Grad-CAM}$

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA

https://arxiv.org/pdf/1610.02391.pdf
http://gradcam.cloudcv.org/
https://github.com/ramprs/grad-cam/

Results

Predicted class	#1 boxer	#2 bull mastiff	#3 tiger cat
Grad-CAM [1]			
Guided backpropagation [2]			
Guided Grad-CAM [1]			

Explaining Machine Learning Models using Entropic Variable Projection

François Bachoc[1], Fabrice Gamboa[1,3], Max Halford[2], Jean-Michel Loubes[1,3] and Laurent Risser[1,3]

[1]Institut de Mathématiques de Toulouse
[2] Institut de recherche en informatique de Toulouse
[3] Artificial and Natural Intelligence Toulouse Institute (3IA ANITI)

https://arxiv.org/pdf/1810.07924.pdf
https://www.gems-ai.com/
https://github.com/XAI-ANITI/ethik

« What-if machine » for group-explainability

Intuition : Re-weighting the observations $\{X_i, Y_i\}_{i=1,\ldots,n}$ to transform a specific property of the test set in average.

Test set

$$\{X_i, Y_i\}_{i=1,\ldots,n}$$

With

$$X_i = \{X_i^1, \ldots, X_i^p\}$$

« Black-box » decision rules

X_i^1 →

X_i^2 →

X_i^3 →

… …

X_i^p →

→ $\hat{Y}_i := f(X_i)$

Explaining Machine Learning Models using Entropic Variable Projection

François Bachoc[1], Fabrice Gamboa[1,3], Max Halford[2], Jean-Michel Loubes[1,3] and Laurent Risser[1,3]

[1]Institut de Mathématiques de Toulouse
[2] Institut de recherche en informatique de Toulouse
[3] Artificial and Natural Intelligence Toulouse Institute (3IA ANITI)

https://arxiv.org/pdf/1810.07924.pdf
https://www.gems-ai.com/
https://github.com/XAI-ANITI/ethik

« What-if machine » for group-explainability

Intuition : Re-weighting the observations $\{X_i, Y_i\}_{i=1,\ldots,n}$ to transform a specific property of the test set in average.

Example (based on the adult income dataset https://www.kaggle.com/uciml/adult-census-income)

Age (X^1)	**Education.num** (X^2)	**Marital.status** (X^3)	**Hours.per.week** (X^4)	…	**Loan granted — True** (Y)	**Loan granted — Predicted** ($\hat{Y} = f_\theta(X)$)
54	4	Divorced	40		No	No
41	10	Never-married	60		Yes	Yes
51	13	Married-civ	40		Yes	No
39	14	Married-civ	65		Yes	Yes
49	10	Divorced	50		No	Yes
…	…	…	…		…	…

3) Three explainability solutions → Entropic Variable Projection

Explaining Machine Learning Models using Entropic Variable Projection

François Bachoc[1], Fabrice Gamboa[1,3], Max Halford[2], Jean-Michel Loubes[1,3] and Laurent Risser[1,3]

[1]Institut de Mathématiques de Toulouse
[2] Institut de recherche en informatique de Toulouse
[3] Artificial and Natural Intelligence Toulouse Institute (3IA ANITI)

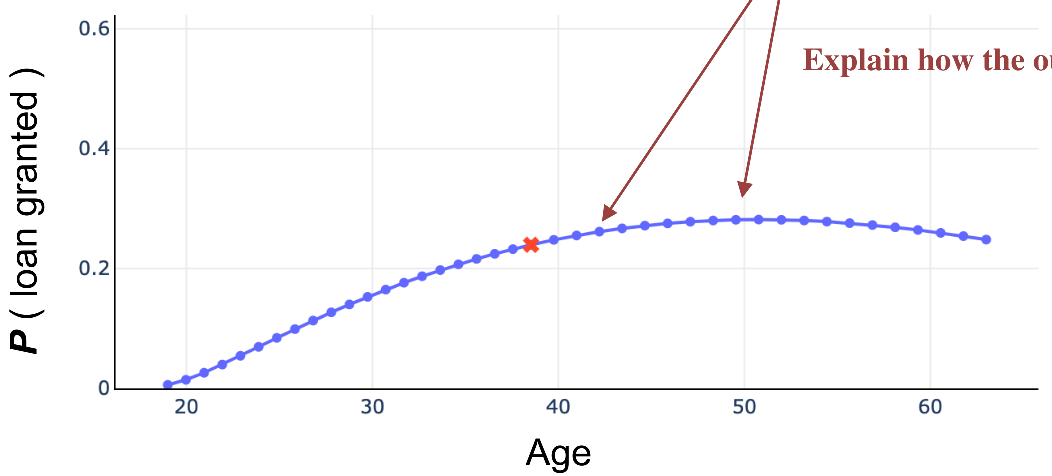
https://arxiv.org/pdf/1810.07924.pdf
https://www.gems-ai.com/
https://github.com/XAI-ANITI/ethik

What-if the average age is 50 instead of 42 in the test set?

Compute optimal weights →

	Age (X^1)	Education.num (X^2)	Marital.status (X^3)	Hours.per.week (X^4)	…	Loan granted — True (Y)	Loan granted — Predicted ($\hat{Y} = f_\theta(X)$)
1.05	54	4	Divorced	40		No	No
0.83	41	10	Never-married	60		Yes	Yes
1.15	51	13	Married-civ	40		Yes	No
0.81	39	14	Married-civ	65		Yes	Yes
1.15	49	10	Divorced	50		No	Yes
…	…	…	…	…		…	…

Explain how the outputs vary



Technical locks addressed in the paper:
- Algorithmic cost in high-dimension
- Risk to test unrealistic observations

Explaining Machine Learning Models using Entropic Variable Projection

François Bachoc[1], Fabrice Gamboa[1,3], Max Halford[2], Jean-Michel Loubes[1,3] and Laurent Risser[1,3]

[1]Institut de Mathématiques de Toulouse
[2] Institut de recherche en informatique de Toulouse
[3] Artificial and Natural Intelligence Toulouse Institute (3IA ANITI)

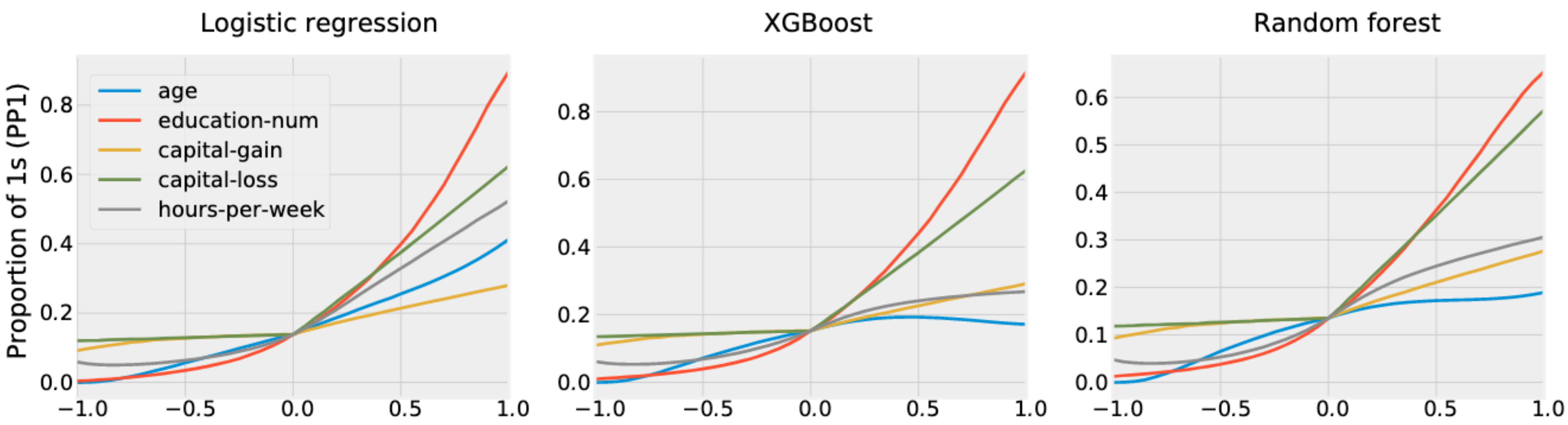
https://arxiv.org/pdf/1810.07924.pdf
https://www.gems-ai.com/
https://github.com/XAI-ANITI/ethik

What-if the average […] is […] instead of [original average value] in the test set?

Compute optimal weights

… then explain

	Age (X^1)	Education.num (X^2)	Marital.status (X^3)	Hours.per.week (X^4)	…	Loan granted — True (Y)	Loan granted — Predicted ($\hat{Y} = f_\theta(X)$)
…	54	4	Divorced	40		No	No
…	41	10	Never-married	60		Yes	Yes
…	51	13	Married-civ	40		Yes	No
…	39	14	Married-civ	65		Yes	Yes
…	49	10	Divorced	50		No	Yes
…	…	…	…	…		…	…



Explaining Machine Learning Models using Entropic Variable Projection

François Bachoc[1], Fabrice Gamboa[1,3], Max Halford[2], Jean-Michel Loubes[1,3] and Laurent Risser[1,3]

[1]Institut de Mathématiques de Toulouse
[2] Institut de recherche en informatique de Toulouse
[3] Artificial and Natural Intelligence Toulouse Institute (3IA ANITI)

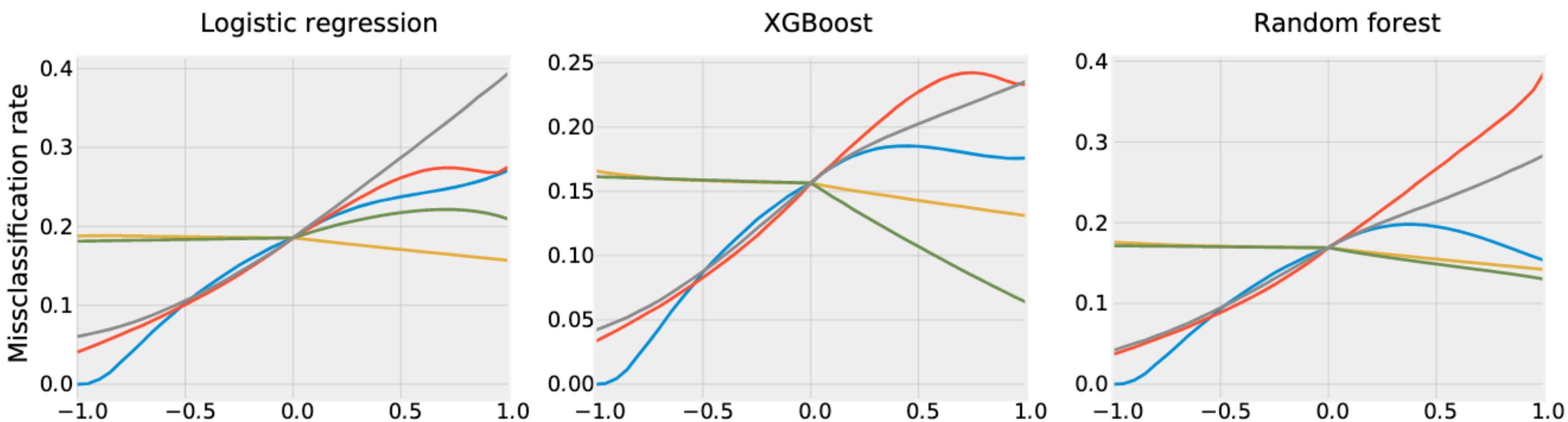
https://arxiv.org/pdf/1810.07924.pdf
https://www.gems-ai.com/
https://github.com/XAI-ANITI/ethik

What-if the average error on […] is […] instead of [original average value] in the test set?

Compute optimal weights →

… then explain

	Age (X^1)	Education.num (X^2)	Marital.status (X^3)	Hours.per.week (X^4)	…	Loan granted — True (Y)	Loan granted — Predicted ($\hat{Y} = f_\theta(X)$)
…	54	4	Divorced	40		No	No
…	41	10	Never-married	60		Yes	Yes
…	51	13	Married-civ	40		Yes	No
…	39	14	Married-civ	65		Yes	Yes
…	49	10	Divorced	50		No	Yes
…	…	…	…	…		…	…



Explaining Machine Learning Models using Entropic Variable Projection

François Bachoc[1], Fabrice Gamboa[1,3], Max Halford[2], Jean-Michel Loubes[1,3] and Laurent Risser[1,3]

[1]Institut de Mathématiques de Toulouse
[2] Institut de recherche en informatique de Toulouse
[3] Artificial and Natural Intelligence Toulouse Institute (3IA ANITI)

https://arxiv.org/pdf/1810.07924.pdf
https://www.gems-ai.com/
https://github.com/XAI-ANITI/ethik

CelebA dataset with a *well-known* bias (http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html)

- >200K celebrity images with 40 binary annotations
- Y_i can be the *Attractive* feature

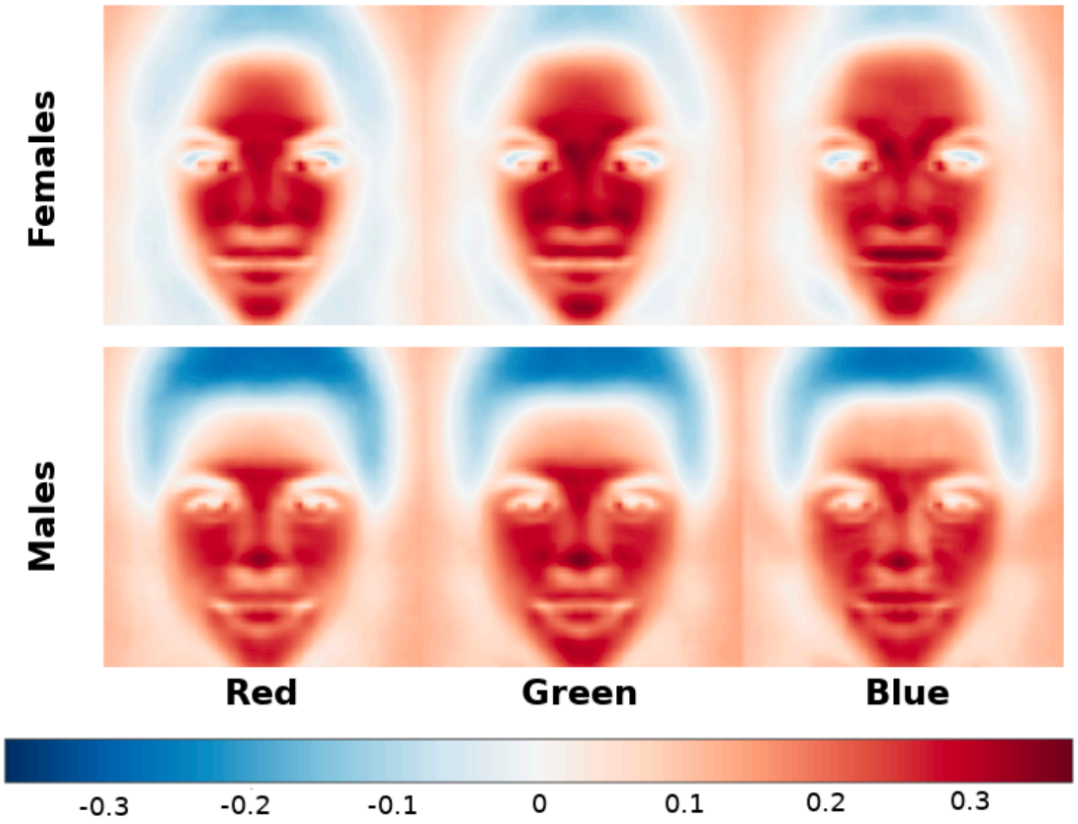
Explaining Machine Learning Models using Entropic Variable Projection

François Bachoc[1], Fabrice Gamboa[1,3], Max Halford[2], Jean-Michel Loubes[1,3] and Laurent Risser[1,3]

[1]Institut de Mathématiques de Toulouse
[2] Institut de recherche en informatique de Toulouse
[3] Artificial and Natural Intelligence Toulouse Institute (3IA ANITI)

https://arxiv.org/pdf/1810.07924.pdf
https://www.gems-ai.com/
https://github.com/XAI-ANITI/ethik

What-if the average prediction of *attractive* is 0.8 instead of [original average value] in the test set?



Average pixel influences to predict whether someone is attractive or not by distinguishing males and females

Conclusion

- Explainability has become an important topic in Machine-Learning.
- Many solution exist, although there are still many open questions (in particular for complex data).
- How to use the outcomes of these explainability techniques to improve the robustness of Black-box predictions or to detect unreliable predictions?

Methodological research on machine learning (M.L.) in Toulouse

3IA ANITI

- 3IA institute gathering 200 researchers in I.A. from Toulouse
- 24 scientific chairs
- 50 industrial partners

Mathematics Institute of Toulouse — IMT

- UMR CNRS, UT3, INSA
- 360 members
- Statistics and Optimisation team working on M.L.

Computer Science Research Institute of Toulouse — IRIT

- UMR CNRS, INPT, UT3, UT1, UT2
- 700 members
- Different teams working on M.L.

Labex CIMI → team A.O.C.

- 27 permanent researchers from IMT and IRIT
- Research in M.L. on broader topics than in 3IA ANITI