Interpretability evaluation framework: Aspects in Machine learning and hydrology domain applications

School of Computer Science and Engineering - Institute of Earth Sciences

Hebrew University

Ronen Rojas

October 2022

Contents

Chapter 1

Abstract

Short overview, write this at the end.

Chapter 2

Introduction

In recent years machine learning models became the ”go-to” solution for almost every task of estimation or prediction. It has Replaced the somewhat tedious task of trying to manually extract insights from the data.

It is definitely easier in a sense to let the computer ”learn” from the data whatever it needs, all you need is a strong GPU and good clean data, once you’ve got all your ducks in a row you’re good to go. The performance of these kind of models is unprecedented and outperform humans on specific tasks ([15], [21], [48], [27], [40]).

But with great power comes great responsibility, how do we explain a model prediction in a sensible manner. Most models are considered to be *”black boxes”* and humanly incomprehensible. Given a machine learning model we are now interested more on ”why” and less on ”what” and most of the time a simple accuracy metric is not enough . This fact gave rise to significant development in the field interpretability methods ([28], [53], [52], [51], [35], [25], [34], [49]).

Interpretability tries to tackle this issue and tries to answer these lines of questions:

- *Why did the model made that prediction?*
- *Do we have a human-friendly explanation for this prediction?*
- *Can we visualize this explanation?*

Some interpretability methods try to tackle the idea of ”why?” instead of ”what?” from an axiomatic approach ([45], [17], [29], [20]), while some do so from a visualization perspective [51]. Even though visualization results can be insightful to humans, such evaluations frameworks are not objectives and time consuming.

Evaluation of interpretability methods is somewhat convoluted and sometimes untrustworthy [46], how does someone define a good explanation? Clearly since there is no ground truth for that there is no right nor wrong, at the end of the day some methods are better then others on different types of models and different types of data-set.

This is the main motivation this research, refinement and exploration of the notion on how can we understand better our interpretability methods and their limitations, how can we evaluate then in a more concrete manner. Can we expand it to a more scientific domain e.g. Hydrology and use some human ground expertise to achieve that.

Chapter 3

Literature review

In this chapter we will review interpretability methods that gained popularity in recent years. We will also review frameworks of evaluating interpretability methods and what evaluation metrics were used. . We'll constrain the scope of this work only to model attribution-based explanations [53], these kind of explanations are local and post-hoc i.e., given a trained model and an input for the prediction, attribution measures or ranks each input feature to explain the model's prediction.

3.1 What is an attribute?

Formally, if we have a model that was trained to perform a specific task:

$$\begin{gathered} y = f_\theta(x) \\ x \in \mathbb{R}^n, y \in \mathbb{R} \end{gathered}$$

An attribution method provides an explanation E for the input x:

$$\begin{gathered} E = \text{attrib}(f_\theta, x) \\ x \in \mathbb{R}^n, E \in \mathbb{R}^n \end{gathered}$$

E gives an explanation power rank for each input coordinate for the prediction $y = f_\theta(x)$. Each coordinate of the input for a model is usually called a feature.

3.2 Interpretability methods

For this section we'll assume we have a model $y = f(x)$ and an attribution-based interpretability method that outputs an explanation E for the prediction y with the same dimensions as the input x.

3.2.1 Grad

The most straight forward interpretability method [8] [41], also known as saliency map:

$$E_{\text{GRAD}}(f, x) = \nabla_x f(x)$$

3.2.2 Input × Gradient

Saliency map can be substantially improved [39] by simply multiplying the gradient with the input, this is basically first-order Taylor approximation in a sense:

$$E_{\text{INPUT}\times\text{GRAD}}(f, x) = x \odot \nabla_x f(x)$$

3.2.3 Integrated Gradients

Sundararajan et al. [45] proposed this interpretability method, abbreviated IG, denote the coordinate i as of input x as x_i:

$$[E_{\text{IG}}(f, x)]_i = (x_i - \overline{x}_i) \times \int_{\alpha=0}^{\alpha=1} \frac{\partial f(x + \alpha(\overline{x} - x))}{\partial x_i} d\alpha$$

where $\overline{x}$ is the baseline input for the method, usually something neutral in the data-set e.g. the black image in images data-sets.

3.2.4 Local Integrated Gradients

Ancona et al. [5] suggested a variant of any method that has an multiplication term by the input. If the aforementioned term is omitted then it's call local method whereas the original is called global, in case of the IG method we'll get:

$$[E_{\text{local-IG}}(f, x)]_i = \int_{\alpha=0}^{\alpha=1} \frac{\partial f(x + \alpha(\overline{x} - x))}{\partial x_i} d\alpha$$

According to [5] *global attribution methods* describe marginal effect of a feature on the output with respect to a baseline, whereas *local attribution methods* describes changes for infinitesimally small perturbations around the original input.

3.2.5 Smooth Grad

Smilkov et al. [42] proposed a adding some noise to the attribution process that will help smooth the saliency maps :

$$E_{\text{SGRAD}}(f,x) = \frac{1}{N}\sum_{i=1}^{N} \nabla_x f(x+g_i)$$

Where g_i are noise vectors $g_i \sim \mathcal{N}(0,\sigma^2)$ are drawn i.i.d. from a normal distribution. One can generalize this method, given an attribution method $E(f,x)$ its smooth counterpart is:

$$E_{\text{Smooth}}(f,x) = \frac{1}{N}\sum_{i=1}^{N} E(f,x+g_i)$$

3.2.6 VarGrad

Adebayo et al. [3] suggests using the variance operator similarly to the SGRAD. obviously this method can be generalized to any attribution method, denote the variance operator $\mathcal{V}$:

$$E_{\text{Variance}}(f,x) = \mathcal{V}\big[E(f,x+g_i)\big]$$

3.2.7 Smooth Grad Square SG-SQ

Hooker et al. [18] proposed the method element-wise square of SGrad

$$E_{\text{SG-SQ}}(f,x) = E_{\text{SGRAD}}(f,x) \odot E_{\text{SGRAD}}(f,x)$$

Where $\odot$ is the element-wise product.

3.2.8 Shapely Value Sampling

Shapley values provide is a concept in game theory to calculate "fair" distribution of winnings for each player members of the winning coalition in a cooperative game by measuring marginal contribution to the final outcome ([10], [43] [37], [26], [1]).

Denote $\mathcal{M}$, the set of players i.e. the grand coalition, $\mathcal{S}$ the set of players in the partial coalition and v is the contribution map between subsets of players, the Shapley value ϕ_i for coalition member i is defined as follows:

$$\phi_i(v) = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{i\}} \binom{|\mathcal{M}|}{1, |\mathcal{S}|, |\mathcal{M}| - |\mathcal{S}| - 1}^{-1} \Big(v(\mathcal{S} \cup \{i\}) - v(\mathcal{S}) \Big)$$

Based on this definition an efficient perturbation process for an attribution-based explanation can be devised [44]. By taking a random permutation of the input features $x_i, i \in [N]$, adding them one-by-one to the given baseline x' and defining $v = f(x) - f(x')$. Repeating this process n times, each time choosing a new random permutation of the input features will yield to an empiric estimation of the real Shapley value.

3.2.9 LRP

Layer-wise relevance propagation abbreviated LRP [22] [29], is a method that tries to estimate every neurons' layer relevancy. The model output score represents the initial relevance which then is being propagated backwards in a set of pre-defined rules the are applied sequentially to all layers of the model. When the propagation reaches the first layer we will get the wanted attribution of the input. This backward propagation mechanism resembles somewhat of a numerical differentiation [52]:

$$[E_{\text{LRP}}(f, x)]_i = x_i \cdot \frac{\partial^g f(x)}{\partial x_i}, \quad g(z) = \frac{\sigma(z)}{z}$$

where $\sigma(z)$ is the non-linearity in the network and $g(z)$ is a replacement the derivative of $\sigma(z)$.

3.2.10 DeepLift

Shrikumar et al. [38] tried to devise a more intricate backward propagation mechanism called Deep Learning Important FeaTures, DeepLift in short.

The mechanism used pre-defined rules for each type of non-linearity and layer type. Their novelty was introducing a reference point in their propagation mechanism.

$$[E_{\text{DeepLift}}(f,x)]_i = (x_i - x_i^{ref}) \cdot \frac{\partial^g f(x)}{\partial x_i}, \quad g(z) = \frac{\sigma(z) - \sigma(z^{ref})}{z - z^{ref}}$$

Both LRP and DeepLIFT have gained the nickname *discrete gradient methods* [52].

3.2.11 Feature Ablation

A perturbation based approach to computing attribution, replacing each input feature x_i with a given baseline x_i', and computing the difference in output.

3.2.12 Guided Grad CAM

Computes element-wise product of guided backpropagation attributions with upsampled (non-negative) GradCAM attributions. GradCAM attributions are computed with respect to the layer provided in the constructor, and attributions are upsampled to match the input size. GradCAM is designed for convolutional neural networks, and is usually applied to the last convolutional layer.

Note that if multiple input tensors are provided, attributions for each input tensor are computed by upsampling the GradCAM attributions to match that input's dimensions. If interpolation is not possible for the input tensor dimensions and interpolation mode, then an empty tensor is returned in the attributions for the corresponding position of that input tensor. This can occur if the input tensor does not have the same number of dimensions as the chosen layer's output or is not either 3D, 4D or 5D.

Note that attributions are only meaningful for input tensors which are spatially alligned with the chosen layer, e.g. an input image tensor for a convolutional layer. Short paragraph about Guided Grad CAM[36]

3.2.13 Deconvolution

Computes attribution using deconvolution. Deconvolution computes the gradient of the target output with respect to the input, but gradients of ReLU functions are overridden so that the gradient of the ReLU input is simply computed taking ReLU of the output gradient, essentially only propagating non-negative gradients (without dependence on the sign of the ReLU input). Short paragraph about Deconvolution[50] [24]

3.2.14 Lime

Short paragraph about Lime [32]

3.3 Interpretability methods frameworks and metrics

3.3.1 Feature importance removal framework

There has been a line of study to treat attribution-based explanations as feature importance ([33] [18] [29],[22], [6], [13], [23], [7], [30]). This line of works asks what will happen if we "remove" these features from the input strategically? what will happen to model predictions?, if we remove from most to least important, ranked from the explanations) we will probably see degradation. This can be evaluated by a certain metric. On the other we can ask the opposite question what will happen if we remove the least important features ? can the model sustain it's merits ? until what percentage of the input is removed ?

By comparing the degradation or robustness of a certain model we can compare different interpretability method attributes, if explanations from one method "broke" the model quicker in a sense this method is better and vice versa if we were able to sustain accuracy or other performance metrics of a model when removing least important features from explanations ranks. Motavon et al. [29] coined this desired property as *Explanation Selectivity*.

Samec et al. [33] have devised an input perturbation process called *most relevant first*, abbreviated MoRF for the trained model $f(x)$. In this process we remove information from the input by setting the coordinates (or a region of surrounding pixels e.g. 9×9 neighborhood) to a uniform distribution. MoRF process is done by the individual attribution ranking e of a given input x, this is done for L iterations:

$$\begin{aligned} & x^{(0)}_{\text{MoRF}} = x \\ \forall 1 \leq k \leq L \quad & x^{(k)}_{\text{MoRF}} = \text{Remove}(x^{(k-1)}_{\text{MoRF}}, e) \end{aligned}$$

The metric used is the area over the MoRF perturbation curve (AOPC):

$$\text{AOPC} = \left\langle \sum_{k=0}^{L} f(x) - f(x^{(k)}_{\text{MoRF}}) \right\rangle_{p(x)}$$

Where $\langle \cdot \rangle_{p(x)}$ means we average over all inputs in the data set, a larger AOPC means a sharper decrease of the MoRF process $(f(x^{(k)}_{\text{MoRF}}))$ thus a better feature importance for the attribution-based explanation.

The opposite approach then is the perturbation process *least relevant first*, abbreviated LeRF. The proposed metric is the area between th perturbations curves:

$$\text{ABPC} = \left\langle \sum_{k=0}^{L} f(x^{(k)}_{\text{LeRF}}) - f(x^{(k)}_{\text{MoRF}}) \right\rangle_{p(x)}$$

In this case the LeRF process needs to sustain the information as much a possibles thus **larger** area is wanted.

In [35] and [22] to evaluate and compare different interpretability method "pixel-flipping" was used. Arras et el. [6] also used LRP evaluation and simply "deleted" word for their model into by setting the corresponding word embedding to zero in order, Lundberg et al. using different strategies on how to remove the input features [23] for evaluations explanations for Tree-based machine learning models.

Hooker et al. [18] opine that any removal of information from the input is disruptive since the metrics will be used on modified input that is not from the data set original distribution, thus the generalization errors of "out of sample" inputs for the model becomes dominant.

To mitigate this case [18] presented a training strategy, RemOve and Retrain (ROAR). This ensures that the metric used, which in the case was a simple test-set accuracy, is used in sample from the distrubution the model was trained on. The opposite approach was to Keep And Retrain (KAR). In this approach, **minimizing** degradation to test-set accuracy is desirable

DeYoung et al. [13] used a similar approach in NLP explanations, called rationales in the NLP jargon, evaluation benchmark. [13] proposed ERASER, which stands for **E**valuation **R**ationales **A**nd **S**imple **E**nglish **R**easoning, a framework that utilized the metric of AOPC (among others metrics) with the only exception that re-training is not necessary since the input size for the model (BERT [12]) can vary so removing tokens (input features) can be done seamlessly.

Chefer et al. [11] used the ERASER framework for the NLP domain and the ”removing” pixels for the vision domain in their evaluation of interpretability methods of self-attentions transformers. In both cases, the metric used was the area-under-the-curve (AUC) perturbation process.

3.3.2 Interpretability as a debug tool framework

Another line of study is to let the interpretability method to be a debug tool for the model or training process itself ([4], [2]). If an interpretability method is sensitive to a model that trained or tested in an ill manner compared to a model that was trained in a nominal way it means that it can better explain the model and in a way could be a sanity check for the interpretability method itself.

Adebayo et al. [4] proposed 2 test frameworks:

1. *Model parameter randomization test*: Apply attributions-based method on a trained model and on the same architecture but on a randomly initialized untrained model.
2. *Data randomization test*: Apply attributions-based method on a trained model and on a copy of the mode trained on a copy of the data set in which all labels were randomly permuted.

Model parameter randomization tests whether attributions method outputs differ substantially between the 2 models, if the output are similar it means the method is insensitive to model parameters which in a sense does not bode well to the explanation goals which are to understand why did the model made its prediction. It's important to note that the randomization of the model parameters was done in a cascading fashion from top to bottom layers.

Data randomization tests again as before the difference between the 2 models, An insensitivity to the permuted labels reveals that the method does not depend on the relationship between input and labels which is not a desired property for an attribution-based method.

[4] used visualization to show the the differences in the attributes, but also used more rigorous ways namely *similarity metrics* as follows:

1. Spearman rank correlation with / without absolute value.
2. Structural similarity index measure (SSIM [47]).
3. Pearson correlation of the histogram of gradients (HOGs)

In his later work Abedeyo et al. [2] used a more elaborate scheme for tampering with the data-set, the training or evaluation process itself, simply named "bugs":

1. *Data contamination bugs*
   (a) *Labeling errors* - Incorrectly labeled data , Similar to previous work.
   (b) *Spurious Correlation* - Make the model associate uncorrelated reason to the task, e.g. blue sky backgrounds with the bird class.
2. *Model contamination bugs* - re-initialization of model weights, similar to previous work.
3. *Test-time contamination bugs* - Out of distribution (OOD) samples, domain shift for the data-set.

Spurious bug implementation was implemented by placing all birds onto one of the sky backgrounds and all dogs onto a bamboo forest background. Logically, explanations on a model that was trained on a data-set like this would identify this correlation i.e. attributing most of the background pixels to the class prediction. In a sense this is a ground truth for the explanation output itself and a "debug" tool for the attribution method.

Test-time contamination bugs assess the ability of attributions to diagnose domain shift, e.g. the attribution for an MNIST input from a model trained on MNIST, to an attribution for the same input but derived from an output of a model trained on a different data-set.

The metric used for the attributes is SSIM to compare the similarity between the explanation in the aforementioned tests.

Heo et al. [16] suggested a quantitative metric in similar spirit, *Fooling Success Rate* (FSR). FSR measures how much an interpretability method is prone in adversarial manipulations in the tested model parameters without changing its accuracy in context of giving sound and reasonable explanations.

3.3.3 Human subject study

Incorporating human study for evaluation of interpretability methods is a very natural ([2], [14], [32]). The explanations coming out of these methods are meant to make black-box machine learning in a sense more understandable and human comprehensible.

Adebayo et al. [2] conduct a 54-person study to assess whether end-users can recognize the bugs in the tests according to the attributions . Evidently this approach showed the people are more biased towards the model predictions even in the presence of its attributions.

Hase et al. [14] tried to test *simulatability* on interpretability methods for machine learning models. A model is simulatable when a person can predict its behavior on new inputs. For this task [14] conducted a two-fold 39-person human-subject study:

1. *Forward simulation* - given an input and an "explanation," users must predict what a model would output for the given input
2. *Counterfactual simulation* - are given an input, a model's output for that input, and an "explanation" of that output, and then they must predict what the model will output when given a perturbation of the original input.

3.3.4 Axiomatic approach framework

In this section we'll present the approach of desired properties which attribution-based interpretability ought to have. These properties are mostly common sense and stem from a more theoretical thought process when devising a scheme for an attribution method hence we call them *axioms* in this context.

Although these *axioms* are not comparable between methods per se they're still worth to mentioned as it can a used as a rigorous framework to evaluate these methods and some can become more quantifiable in a sense as we'll explain later in this section.

3.3.4.1 Sensitivity, implementation invariance, linearity and Symmetry preserving

Sundararajan et al. [45] introduced these axioms in his novel attributed-based method *integrated gradients*:

- *Sensitivity* - If the function implemented by the deep network does not depend (mathematically) on some variable, then the attribution to that variable is always zero.
- *Implementation invariance* - Attributions should be identical for two functionally equivalent networks.
- *Linearity* - Attributions method should preserve any linearity within the network.
- *Symmetry preserving* - If 2 inputs to the network are symmetrical $F(x, y) = F(y, x)$, so should be their corresponding attributions.

3.3.4.2 Continuity

Montavon et al [29] introduced the desired axiomatic property of *continuity* as follows:

- *Continuity* - If two data points are nearly equivalent, then the explanations of their predictions should also be nearly equivalent.

This axiom can be also quantified as follows, denote an attribution method E, two inputs x, x' and their explanations respectively E, E' :

$$\max_{x \neq x'} \frac{\|E - E'\|_1}{\|x - x'\|_2}$$

Bhatt et al. [9] used a similar definition but used the term *low/average sensitivity*

3.3.4.3 Input invariance

Kindermans et al. [20] introduced the *Input invariance* axiom:

- *Input invariance* - The attribution-based method should mirror the sensitivity of the model with respect to transformations of the input

For example a constant shift in the input with two model that were trained on the original data and the shifted data and have the **same predictions** should have the **same attributions**.

3.3.4.4 Monotonicity

Although *monotonicity* defined as metric in [31] it stems from an axiomatic approach on the mechanism of attribution methods that assign an importance value to each feature. From a more theoretical approach a more concrete metric was proposed namely *monotonicity.*

Nguyen et al. [31] posit that feature importance rank attributions should follow a desired property, denote the explanation E_i for feature $i \in [N], y^* = f(\mathbf{x}^*)$:

$$|E_i| \propto \mathbb{E}(l(y^*, f_i)|\mathbf{x}^*_{-i}) = \int_{\mathcal{X}_i} l(y^*, f_i(x_i)p(x_i)dx_i$$

Where l is the loss function, density probability function for feature x_i is $p(x_i)$ and f_i is the restriction of the function f to the feature i obtained by fixing the other features at the values $\mathbf{x}^*_{-i} = (x_1, ..., x_{i-1}, x_{i+1}, ..., x_N)$, so the monotonicity metric for attribution E is defined as the Spearman's correlation coefficient ρ_S:

$$\begin{aligned}
&\mathbf{e} = [|E_1|, ..., |E_i|, ..., |E_N|]\\
&\mathbf{f} = [\mathbb{E}(l(y^*, f_1)|\mathbf{x}^*_{-1}), ..., \mathbb{E}(l(y^*, f_i)|\mathbf{x}^*_{-i}), ..., \mathbb{E}(l(y^*, f_N)|\mathbf{x}^*_{-N})]\\
&\text{monotonicity}(E, f, \mathbf{x}^*, y^*) = \rho_S(\mathbf{e}, \mathbf{f})
\end{aligned}$$

3.3.4.5 Completeness and sensitivity-n

Sundararajan et al. [45] proved that the attribution-based method *itegrated gradients* holds the axiom of *completeness*:

$$\sum_{i=1}^{N} IG(\mathbf{x}) = f(\mathbf{x}) - f(\mathbf{x}')$$

Where $\mathbf{x}'$ is a baseline input that is used for the attribution calculation and we'll be elaborated in later sections.

Ancona et al. [5] generalized the *completeness* axioms namely, *sensitivity-n*:

- *sensitivity-n*: For any subset of features of cardinality n , $\mathbf{x}_s = [x_1, ..., x_n] \subseteq \mathbf{x}$ it holds $\sum_{i=1}^{n} E_i = f(\mathbf{x}) - f(\mathbf{x}_{[\mathbf{x}_S=0]})$

It easy to see that *completeness* axiom is a private case of the *sensitivity-n* axiom when $n = N$. Bhatt et al.

[9] introduced a similar notion of *faithfulness* where the sum of the attributions and the difference in output when setting those features to a reference baseline should be proportionality correlated

3.4 Recognized gap

There seems to be a knowledge gap on how to effectively evaluate and validate any given interpretability method. elaborate more

Chapter 4

Research objective

We offer a framework for evaluating an interpretability method. Our approach can be summarized in the following scheme 4.1:

1. Generate a 2-modal data-set
2. Train a machine learning model on the data-set
3. ‘Train two classifiers for the modalities of the data-set:
   (a) A clean view – training only with the data-set
   (b) An interpreted view – training only with attributes of the trained model
4. Evaluate interpretability method by comparison of the clean classifier to interpreted classifier with simple classification metrics

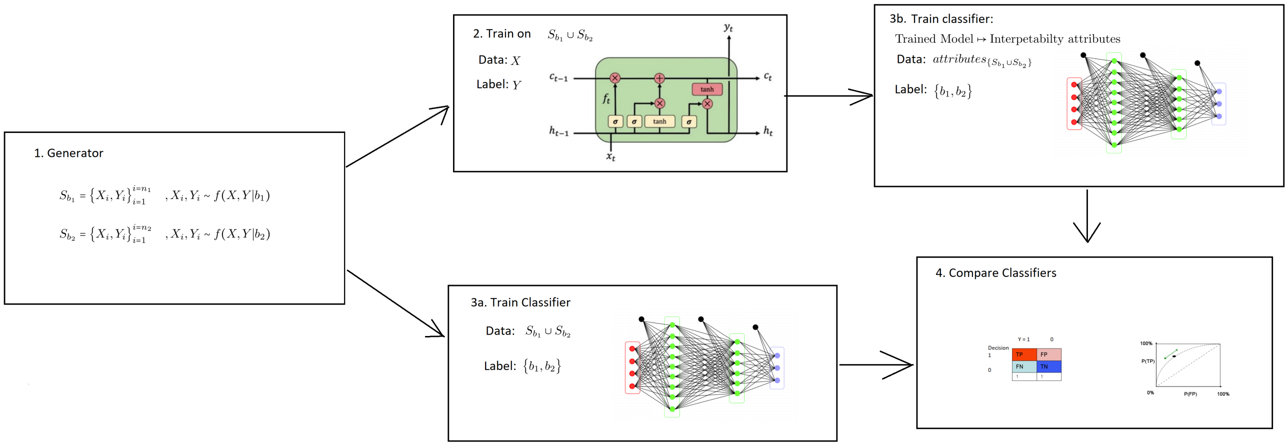
1. Generator
$S_{b_1} = \{X_i, Y_i\}_{i=1}^{i=n_1} \quad , X_i, Y_i \sim f(X, Y|b_1)$
$S_{b_2} = \{X_i, Y_i\}_{i=1}^{i=n_2} \quad , X_i, Y_i \sim f(X, Y|b_2)$
2. Train on $S_{b_1} \cup S_{b_2}$
Data: X
Label: Y
y_t
c_{t-1}
c_t
f_t
tanh
σ
σ
tanh
σ
h_{t-1}
h_t
x_t
3b. Train classifier:
Trained Model $\mapsto$ Interpetabilty attributes
Data: $attributes_{\{S_{b_1} \cup S_{b_2}\}}$
Label: $\{b_1, b_2\}$
3a. Train Classifier
Data: $S_{b_1} \cup S_{b_2}$
Label: $\{b_1, b_2\}$
4. Compare Classifiers
Y = 1
0
Decision
1
0
TP
FP
FN
TN
1
1
100%
P(TP)
0%
P(FP)
100%

Figure 4.1: Scheme

Chapter 5

Data and Models

5.1 Data

5.1.1 DREAM

5.1.2 CARAVAN

5.2 Models

5.2.1 Feed Forward Neural Network

5.2.1.1 Neural Network - Introduction

- $y = f_W(x), x \in \mathbb{R}^n, y \in \mathbb{R}^m$
- W Learn-able parameters
- Data set $\{x_i, y_i\}_{i=1}^k$
- Loss function $L(f(x_i), y_i)$
- Gradient decent $W_t = W_{t-1} + \alpha \nabla_W L(x, y)$

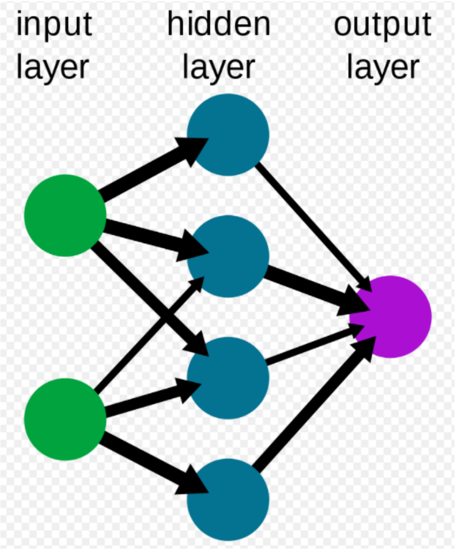

Figure 5.1: Neural Network Architecture

5.2.1.2 Neural Network - Function definitions

- Affine layer $y = Ax + b, x \in \mathbb{R}^n, b, y \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$
- Activation point wise
  - $\sigma_g(z) = \frac{1}{1+e^{-z}}$ - Sigmoid
  - $ReLU(z) = \max(0, z)$ - Rectified linear unit

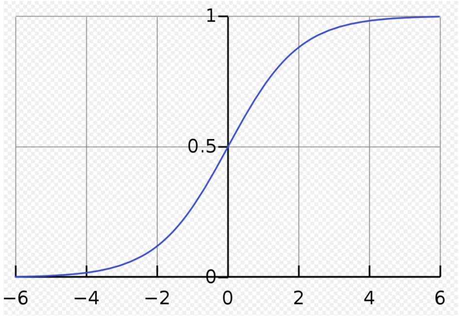

Figure 5.2: Sigmoid function

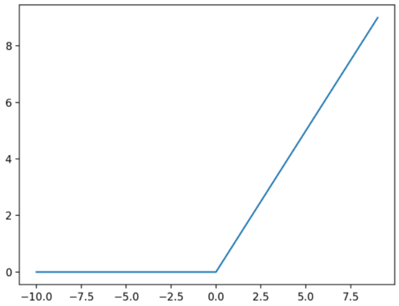

Figure 5.3: ReLU function

5.2.1.3 Neural Network - Layering functions

- Dense: $y = \sigma(L), L = Ax + b$
- Expressiveness
- Back propagation - $\frac{\partial y}{\partial w} = \frac{\partial y}{\partial L} \cdot \frac{\partial L}{\partial w}$

5.2.1.4 Neural Network - Hyperbolic tangent function

$\sigma_h = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

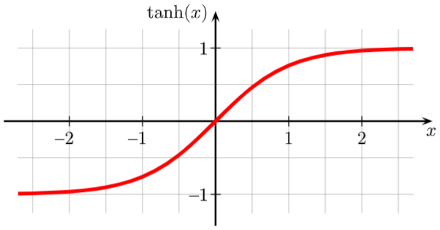

Figure 5.4: Hyperbolic tangent function

5.2.2 RNN

5.2.2.1 RNN - Introduction

- Signals with timestamp (time series) - $\{x_t, y_t\}_{t=1}^{k}$
- Hidden state - $h_t = f_W(x_t, h_{t-1})$
- Same weights W for each step
- Popular in Natural Language Processing (NLP)

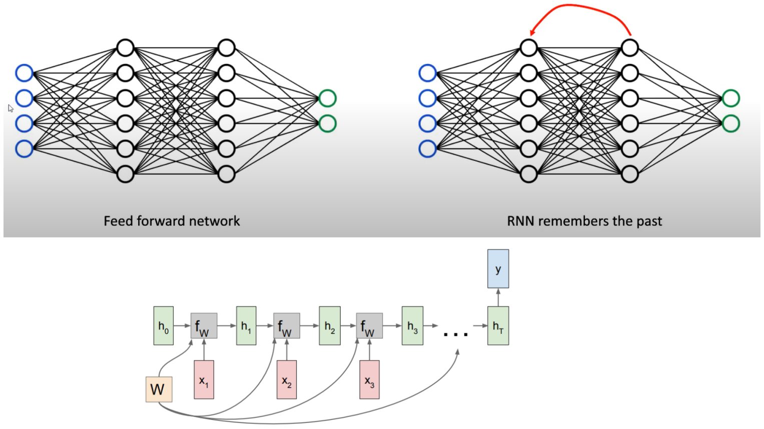

Figure 5.5: RNN

5.2.2.2 RNN - Multi-Layer

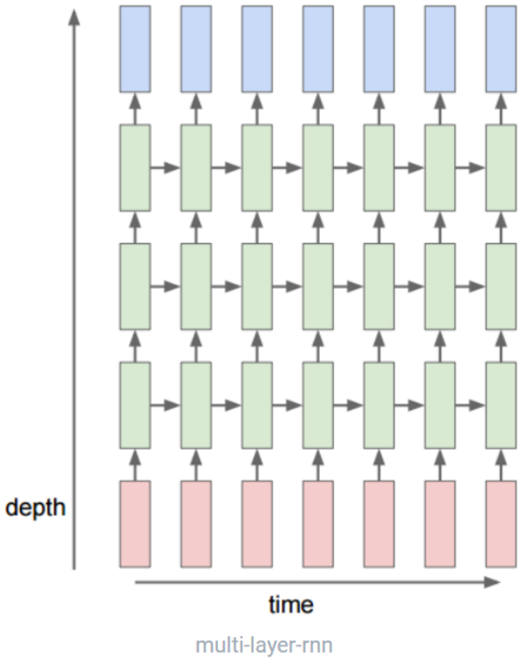

Figure 5.6: RNN - MultiLayer

$$h_t^{layer} = \sigma_h\left(W^{layer}\begin{pmatrix} h_t^{layer-1} \\ h_{t-1}^{layer} \end{pmatrix}\right)$$

5.2.3 LSTM

5.2.3.1 LSTM - Motivation

LSTM (Long Short Term Memory) is a special kind of RNN, designed to overcome the limitation of RNN

- Gradient vanishing and exploding
- Complex training
- Difficulty to processes long sequences

Remembering information for long periods of time is intrinsic to LSTM.

5.2.3.2 LSTM - Principles

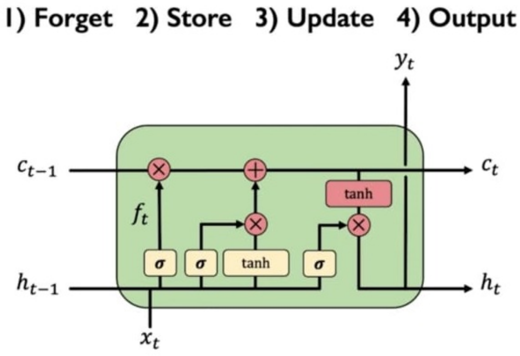

Figure 5.7: LSTM - Scheme

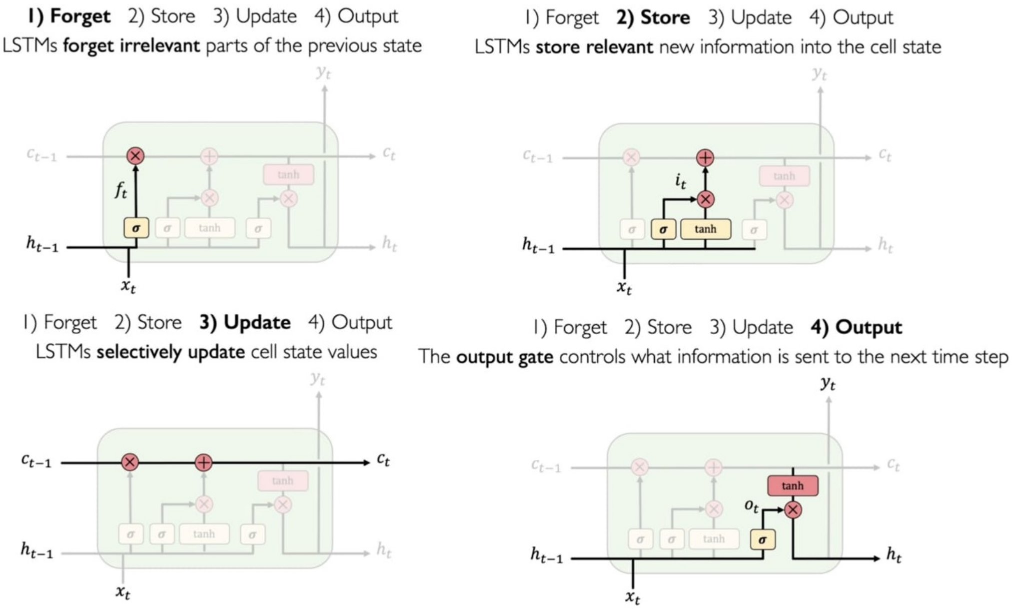
I) Forget 2) Store 3) Update 4) Output
LSTMs forget irrelevant parts of the previous state
c_{t-1}
c_t
y_t
f_t
σ
tanh
h_{t-1}
h_t
x_t
I) Forget 2) Store 3) Update 4) Output
LSTMs store relevant new information into the cell state
i_t
I) Forget 2) Store 3) Update 4) Output
LSTMs selectively update cell state values
I) Forget 2) Store 3) Update 4) Output
The output gate controls what information is sent to the next time step
o_t

Figure 5.8: LSTM - Scheme

- Separate Cell state
- Gate to control flow of information:
    - **Forget** - Gets rids of irrelevant information.
    - **Store** - Relevant information from input
    - **Update** - Selectively update cell state
    - **Output**

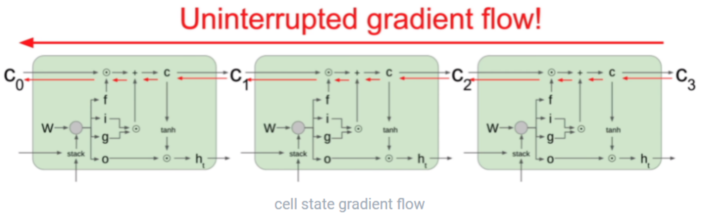

Figure 5.9: LSTM - Gradient flow

5.2.3.3 LSTM- Formulation

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

where the initial values are $c_0 = 0$ and $h_0 = 0$ and the operator $\circ$ denotes the Hadamard product (element-wise product).

- $x_t \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in \mathbb{R}^h$: forget gate's activation vector
- $i_t \in \mathbb{R}^h$: input/update gate's activation vector
- $o_t \in \mathbb{R}^h$: output gate's activation vector
- $h_t \in \mathbb{R}^h$: hidden state vector also known as output vector of the LSTM unit
- $\tilde{c}_t \in \mathbb{R}^h$: cell input activation vector
- $c_t \in \mathbb{R}^h$: cell state vector
- $W \in \mathbb{R}^{h\times d}, U \in \mathbb{R}^{h\times h}$ and $b \in \mathbb{R}^h$: weight matrices and bias vector parameters which need to be learned during training

where the superscripts d and h refer to the number of input features and number of hidden units, respectively.

5.2.3.4 LSTM- Multi-cell

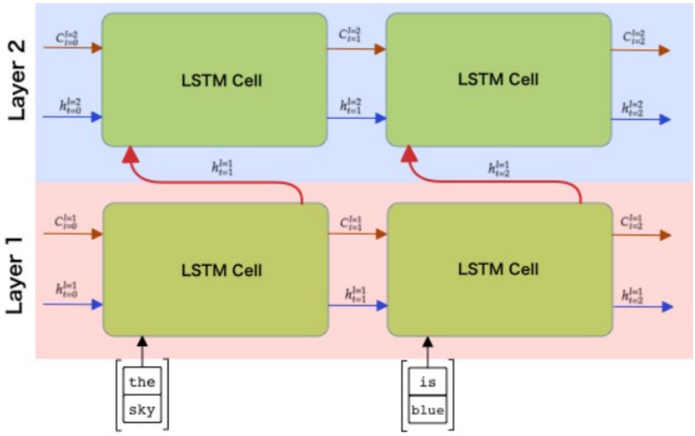

Figure 5.10: LSTM - Multi layer cells

5.2.3.5 LSTM - in context of rainfall

- 2018
  - CAMELS - Data set
  - 2-Layer LSTM -cells
  - Input feature size $d = 5$ - prcp(mm/day), srad(W/m2), tmax(C), tmin(C), vp(Pa).
  - Hidden state size $h = 20$
  - Dropout rate = 10%
  - Sequemce length - 365 days.
- 2019
  - CAMELS - Data set
  - 1-Layer LSTM
  - Input feature size $d = 5$?
  - Hidden state size $h = 256$
  - Dropout rate = 40%
  - Sequence length - 270 days.

Chapter 6

Results

6.1 Sanity Check

6.2 Dream synthetic model

6.3 CelebA data set

6.4 Caravan Data set

Chapter 7

Discussions

Chapter 8

Conclusions

Maybe write about Integrated Hessians [19]

Bibliography

[1] Kjersti Aas, Martin Jullum, and Anders Løland. *Explaining individual predictions when features are dependent: More accurate approximations to Shapley values*. 2019. DOI: 10.48550/ARXIV.1903.10464. URL: https://arxiv.org/abs/1903.10464.

[2] Julius Adebayo et al. "Debugging Tests for Model Explanations". In: *CoRR* abs/2011.05429 (2020). arXiv: 2011.05429. URL: https://arxiv.org/abs/2011.05429.

[3] Julius Adebayo et al. "Local Explanation Methods for Deep Neural Networks Lack Sensitivity to Parameter Values". In: *CoRR* abs/1810.03307 (2018). arXiv: 1810.03307. URL: http://arxiv.org/abs/1810.03307.

[4] Julius Adebayo et al. *Sanity Checks for Saliency Maps*. 2018. DOI: 10.48550/ARXIV.1810.03292. URL: https://arxiv.org/abs/1810.03292.

[5] Marco Ancona et al. *Towards better understanding of gradient-based attribution methods for Deep Neural Networks*. 2017. DOI: 10.48550/ARXIV.1711.06104. URL: https://arxiv.org/abs/1711.06104.

[6] Leila Arras et al. ""What is Relevant in a Text Document?": An Interpretable Machine Learning Approach". In: *CoRR* abs/1612.07843 (2016). arXiv: 1612.07843. URL: http://arxiv.org/abs/1612.07843.

[7] Leila Arras et al. "Evaluating Recurrent Neural Network Explanations". In: *CoRR* abs/1904.11829 (2019). arXiv: 1904.11829. URL: http://arxiv.org/abs/1904.11829.

[8] David Baehrens et al. “How to explain individual classification decisions”. In: *The Journal of Machine Learning Research* 11 (2010), pp. 1803–1831.

[9] Umang Bhatt, Adrian Weller, and Jose M. F. Mour. “Evaluating and Aggregating Feature-based Model Explanations”. In: *CoRR* abs/2005.00631 (2020). arXiv: 2005.00631. URL: https://arxiv.org/abs/2005.00631.

[10] Javier Castro, Daniel Gómez, and Juan Tejada. “Polynomial calculation of the Shapley value based on sampling”. In: *Computers and Operations Research* 36.5 (2009). Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X), pp. 1726–1730. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2008.04.004. URL: https://www.sciencedirect.com/science/article/pii/S0305054808000804.

[11] Hila Chefer, Shir Gur, and Lior Wolf. “Transformer Interpretability Beyond Attention Visualization”. In: *CoRR* abs/2012.09838 (2020). arXiv: 2012.09838. URL: https://arxiv.org/abs/2012.09838.

[12] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[13] Jay DeYoung et al. “ERASER: A Benchmark to Evaluate Rationalized NLP Models”. In: *CoRR* abs/1911.03429 (2019). arXiv: 1911.03429. URL: http://arxiv.org/abs/1911.03429.

[14] Peter Hase and Mohit Bansal. “Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?” In: *CoRR* abs/2005.01831 (2020). arXiv: 2005.01831. URL: https://arxiv.org/abs/2005.01831.

[15] Kaiming He et al. *Deep Residual Learning for Image Recognition.* 2015. arXiv: 1512.03385 [cs.CV].

[16] Juyeon Heo, Sunghwan Joo, and Taesup Moon. "Fooling Neural Network Interpretations via Adversarial Model Manipulation". In: *CoRR* abs/1902.02041 (2019). arXiv: 1902.02041. URL: http://arxiv.org/abs/1902.02041.

[17] Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. "Fast Axiomatic Attribution for Neural Networks". In: *CoRR* abs/2111.07668 (2021). arXiv: 2111.07668. URL: https://arxiv.org/abs/2111.07668.

[18] Sara Hooker et al. *A Benchmark for Interpretability Methods in Deep Neural Networks*. 2018. DOI: 10.48550/ARXIV.1806.10758. URL: https://arxiv.org/abs/1806.10758.

[19] Joseph D. Janizek, Pascal Sturmfels, and Su-In Lee. "Explaining Explanations: Axiomatic Feature Interactions for Deep Networks". In: *CoRR* abs/2002.04138 (2020). arXiv: 2002.04138. URL: https://arxiv.org/abs/2002.04138.

[20] Pieter-Jan Kindermans et al. *The (Un)reliability of saliency methods*. 2017. DOI: 10.48550/ARXIV.1711.00867. URL: https://arxiv.org/abs/1711.00867.

[21] Kamran Kowsari et al. "RMDL: Random Multimodel Deep Learning for Classification". In: *CoRR* abs/1805.01890 (2018). arXiv: 1805.01890. URL: http://arxiv.org/abs/1805.01890.

[22] Sebastian Lapuschkin et al. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLoS ONE* 10 (July 2015), e0130140. DOI: 10.1371/journal.pone.0130140.

[23] Scott M. Lundberg et al. "Explainable AI for Trees: From Local Explanations to Global Understanding". In: *CoRR* abs/1905.04610 (2019). arXiv: 1905.04610. URL: http://arxiv.org/abs/1905.04610.

[24] Aravindh Mahendran and Andrea Vedaldi. "Salient Deconvolutional Networks". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 120–135. ISBN: 978-3-319-46466-4.

[25] Aniek F. Markus, Jan A. Kors, and Peter R. Rijnbeek. “The role of explainability in creating trustworthy artificial intelligence for health care: a comprehensive survey of the terminology, design choices, and evaluation strategies”. In: *CoRR* abs/2007.15911 (2020). arXiv: 2007.15911. URL: https://arxiv.org/abs/2007.15911.

[26] Rory Mitchell et al. *Sampling Permutations for Shapley Value Estimation.* 2021. DOI: 10.48550/ARXIV.2104.12199. URL: https://arxiv.org/abs/2104.12199.

[27] Volodymyr Mnih et al. “Playing Atari with Deep Reinforcement Learning”. In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[28] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable.* https://christophm.github.io/interpretable-ml-book/. 2019.

[29] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. “Methods for Interpreting and Understanding Deep Neural Networks”. In: *CoRR* abs/1706.07979 (2017). arXiv: 1706.07979. URL: http://arxiv.org/abs/1706.07979.

[30] Dong Nguyen. “Comparing Automatic and Human Evaluation of Local Explanations for Text Classification”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1069–1078. DOI: 10.18653/v1/N18-1097. URL: https://aclanthology.org/N18-1097.

[31] An-phi Nguyen and María Rodríguez Martínez. “On quantitative aspects of model interpretability”. In: *CoRR* abs/2007.07584 (2020). arXiv: 2007.07584. URL: https://arxiv.org/abs/2007.07584.

[32] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “”Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *CoRR* abs/1602.04938 (2016). arXiv: 1602.04938. URL: http://arxiv.org/abs/1602.04938.

[33] Wojciech Samek et al. *Evaluating the visualization of what a Deep Neural Network has learned.* 2015. DOI: 10.48550/ARXIV.1509.06321. URL: https://arxiv.org/abs/1509.06321.

[34] Wojciech Samek et al. *Explainable AI: interpreting, explaining and visualizing deep learning*. Vol. 11700. Springer Nature, 2019.

[35] Wojciech Samek et al. “Toward Interpretable Machine Learning: Transparent Deep Neural Networks and Beyond”. In: *CoRR* abs/2003.07631 (2020). arXiv: 2003.07631. URL: https://arxiv.org/abs/2003.07631.

[36] Ramprasaath R. Selvaraju et al. “Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization”. In: *CoRR* abs/1610.02391 (2016). arXiv: 1610.02391. URL: http://arxiv.org/abs/1610.02391.

[37] L. S. Shapley. “17. A Value for n-Person Games”. In: *Contributions to the Theory of Games (AM-28), Volume II*. Ed. by Harold William Kuhn and Albert William Tucker. Princeton: Princeton University Press, 2016, pp. 307–318. ISBN: 9781400881970. DOI: doi:10.1515/9781400881970-018. URL: https://doi.org/10.1515/9781400881970-018.

[38] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning Important Features Through Propagating Activation Differences”. In: *CoRR* abs/1704.02685 (2017). arXiv: 1704.02685. URL: http://arxiv.org/abs/1704.02685.

[39] Avanti Shrikumar et al. “Not Just a Black Box: Learning Important Features Through Propagating Activation Differences”. In: *CoRR* abs/1605.01713 (2016). arXiv: 1605.01713. URL: http://arxiv.org/abs/1605.01713.

[40] David Silver et al. “Mastering the Game of Go with Deep Neural Networks and Tree Search”. In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. DOI: 10.1038/nature16961.

[41] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2013. DOI: 10.48550/ARXIV.1312.6034. URL: https://arxiv.org/abs/1312.6034.

[42] Daniel Smilkov et al. “SmoothGrad: removing noise by adding noise”. In: *CoRR* abs/1706.03825 (2017). arXiv: 1706.03825. URL: http://arxiv.org/abs/1706.03825.

[43] Erik Strumbelj and Igor Kononenko. "An Efficient Explanation of Individual Classifications using Game Theory". In: *J. Mach. Learn. Res.* 11 (2010), pp. 1–18.

[44] Erik Strumbelj and Igor Kononenko. "An efficient explanation of individual classifications using game theory". In: *The Journal of Machine Learning Research* 11 (2010), pp. 1–18.

[45] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *CoRR* abs/1703.01365 (2017). arXiv: 1703.01365. URL: http://arxiv.org/abs/1703.01365.

[46] Richard Tomsett et al. "Sanity Checks for Saliency Metrics". In: *CoRR* abs/1912.01451 (2019). arXiv: 1912.01451. URL: http://arxiv.org/abs/1912.01451.

[47] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[48] Martin Wistuba, Ambrish Rawat, and Tejaswini Pedapati. "A Survey on Neural Architecture Search". In: *CoRR* abs/1905.01392 (2019). arXiv: 1905.01392. URL: http://arxiv.org/abs/1905.01392.

[49] Hao Yuan et al. "Explainability in Graph Neural Networks: A Taxonomic Survey". In: *CoRR* abs/2012.15445 (2020). arXiv: 2012.15445. URL: https://arxiv.org/abs/2012.15445.

[50] Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *CoRR* abs/1311.2901 (2013). arXiv: 1311.2901. URL: http://arxiv.org/abs/1311.2901.

[51] Quanshi Zhang and Song-Chun Zhu. "Visual Interpretability for Deep Learning: a Survey". In: *CoRR* abs/1802.00614 (2018). arXiv: 1802.00614. URL: http://arxiv.org/abs/1802.00614.

[52] Yu Zhang et al. "A Survey on Neural Network Interpretability". In: *CoRR* abs/2012.14261 (2020). arXiv: 2012.14261. URL: https://arxiv.org/abs/2012.14261.

[53] Jianlong Zhou et al. "Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics". In: *Electronics* 10.5 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10050593. URL: https://www.mdpi.com/2079-9292/10/5/593.