# 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

August 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 uncanny and unprecedented ([He+15], [Kow+18], [WRP19]).

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 [Mol19].

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 [STY17], [HSR21], while some do so from a visualization perspective [ZZ18]. Evaluation of interpretability methods is somewhat tricky in this sense , how does someone define a good explanation ? at the end some work better then others on different types of models and different types of data-set. This actually motivated use to refine and explore this notion.

# Chapter 3

# Literature review

In this chapter we'll review recent work for frameworks of evaluating interpretability methods and what data-sets were used. We're also going to review the interpretability method used in the current framework

## 3.1 Interpretability methods frameworks and metrics

There has been a line of study to treat explains of interpretability methods a.k.a attributes as feature importance [Sam+15] [Hoo+18]. These works ask what will happen if we "remove" these features from the input strategically? what will happen to model predictions now?, if we remove from most to least important 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" taking out the most important feature the model quicker in a sense this method is better and vice versa if we took the least important features.

Paragraph about the methods ROAR , KAR, MORF, LEVF

Paragraph about the metrics AOPC, SSIM

Another line of study is let interpretability method be a debug tool for the model itself [Ade+18b], [Ade+20]. This line of work studies how sensitive an interpretability method is to contamination of the learning process of a model. If an interpretability method is sensitive to a model that trained or tested in an ill manner compered 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.

Paragraph about the modes were the model was disrupted

Paragraph about the metrics used

## 3.2 Interpretability methods

### 3.2.1 Grad

short paragraph about method [Bae+10], [SVZ13]

### 3.2.2 Smooth Grad

Paragraph about method [Smi+17]

### 3.2.3 VarGrad

Paragraph about method [Ade+18a]

### 3.2.4 Smooth Grad Square SG-SQ

Paragraph about method [Hoo+18]

### 3.2.5 Integrated Gradients

Paragraph about method [STY17]

### 3.2.6 Local Integrated Gradients

Paragraph about method [Anc+17]

### 3.2.7 Shapely Value Sampling

Paragraph about method [CGT09], [SK10]

### 3.2.8 Deep Lift SHAP

Paragraph about method [LL17]

### 3.2.9 local Deep lift SHAP

Paragraph about method [Anc+17]

### 3.2.10 Feature Ablation

"A perturbation based approach to computing attribution, involving replacing each input feature with a given baseline / reference, and computing the difference in output. By default, each scalar value within each input tensor is taken as a feature and replaced independently. Passing a feature mask, allows grouping features to be ablated together. This can be used in cases such as images, where an entire segment or region can be ablated, measuring the importance of the segment (feature group). Each input scalar in the group will be given the same attribution value equal to the change in target as a result of ablating the entire feature group."

## 3.3 Recognized gap

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

# Chapter 4

# Research objective

We offer a framework for evaluating an interpretability method. Our approach can be summarized in the following scheme [Figure1]: 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

# Chapter 5

# Methods

## 5.1 Data

### 5.1.1 DREAM

### 5.1.2 CARAVAN

## 5.2 Models

### 5.2.1 Feed Forward Neural Network

**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 $W_t = W_{t-1} + \alpha \nabla_W L(x, y)$

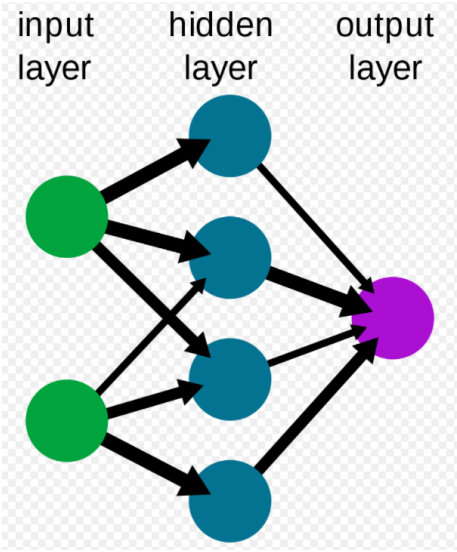


Figure 5.1: Neural Network Architecture

**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

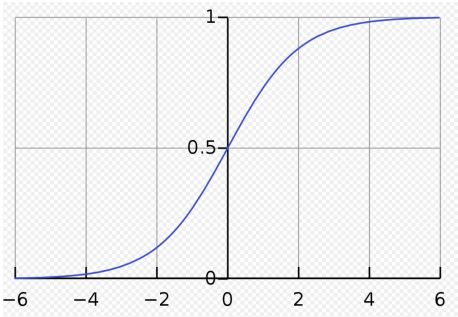


Figure 5.2: Sigmoid function

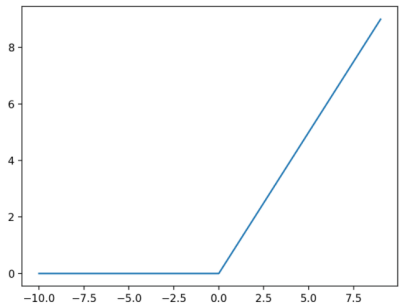


Figure 5.3: ReLU function

**Neural Network - Layering functions**

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

**Neural Network - Hyperbolic tangent function**

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

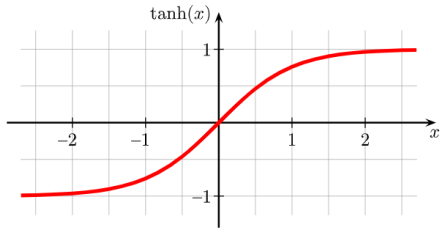


Figure 5.4: Hyperbolic tangent function

**Neural Network - Problems**

- Fixed size input
- No memory

### 5.2.2 RNN

#### 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)

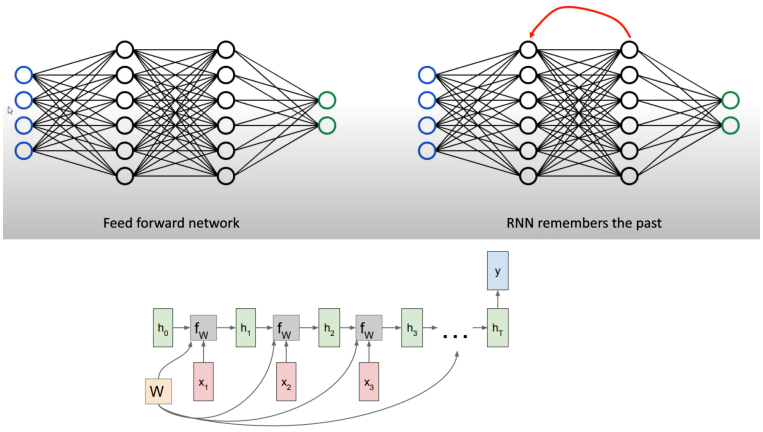


Figure 5.5: RNN

#### RNN - Simple example

- $h_t = \sigma_h\left(W_{hh}h_{t-1} + W_{xh}x_t + b_h\right)$
- $y_t = W_{hy}h_t + b_y$

**RNN - Multi-Layer**

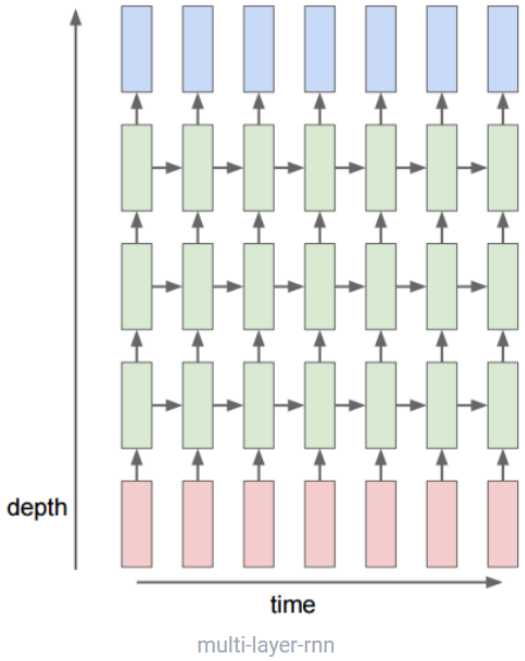


Figure 5.6: RNN - MultiLayer

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

**RNN - Training**

Back propagation through time:

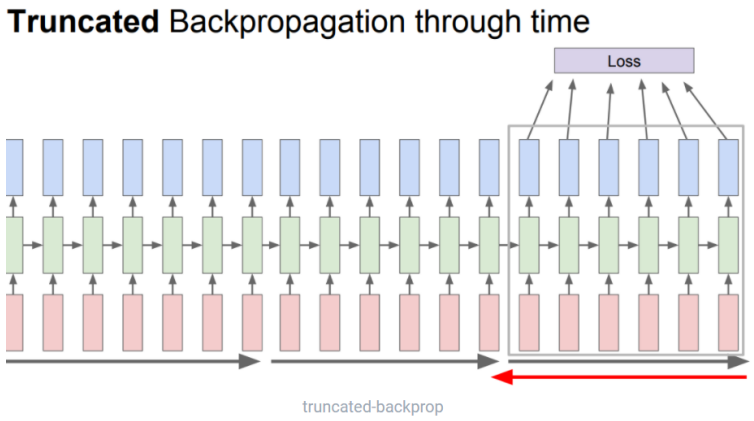


Figure 5.7: RNN - back propagation

### RNN - Problems

Small example, scalars and no hidden states:

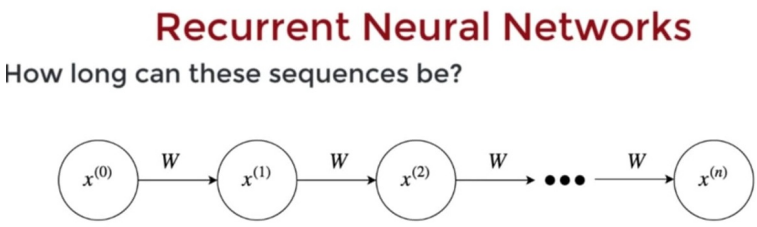


Figure 5.8: RNN - Exploding vanishing gradients

- Exploding vanishing gradients
- $x^{(n)} = W^t x^{(0)}$
- 10 time steps.

### 5.2.3 LSTM

#### 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.

#### LSTM - Principles

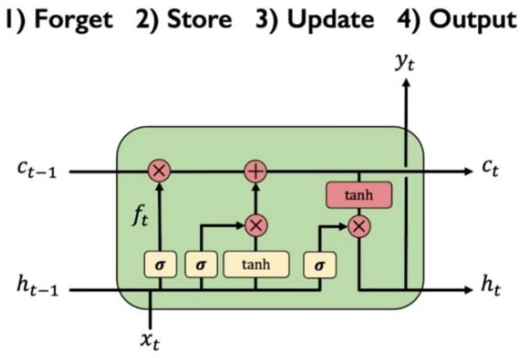


Figure 5.9: LSTM - Scheme

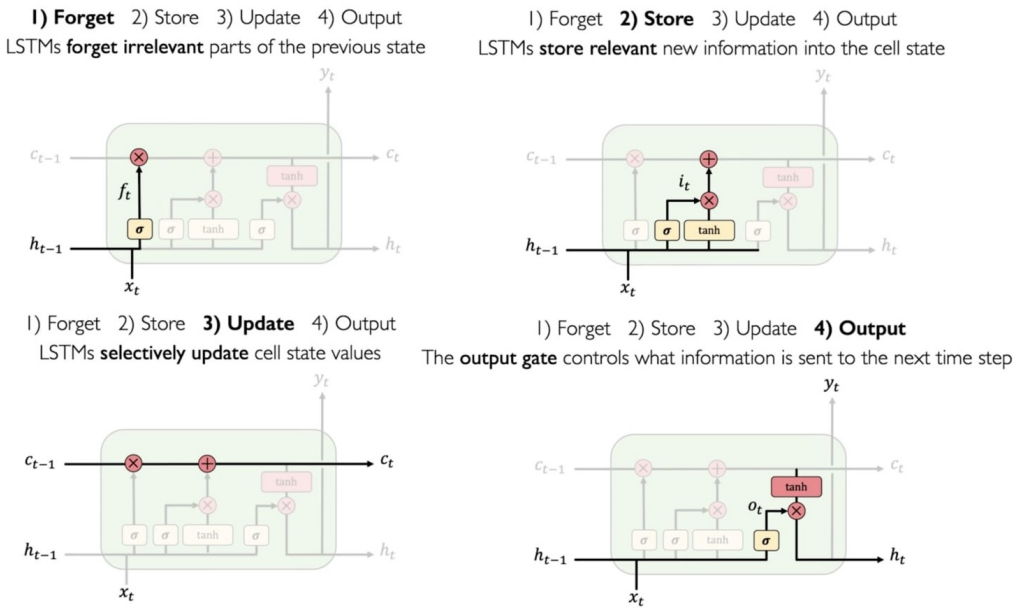


Figure 5.10: LSTM - Scheme

- 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**

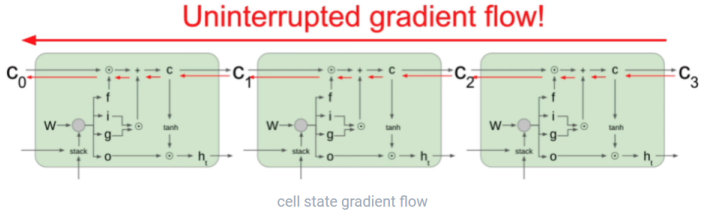


Figure 5.11: LSTM - Gradient flow

**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.

### LSTM- Multi-cell

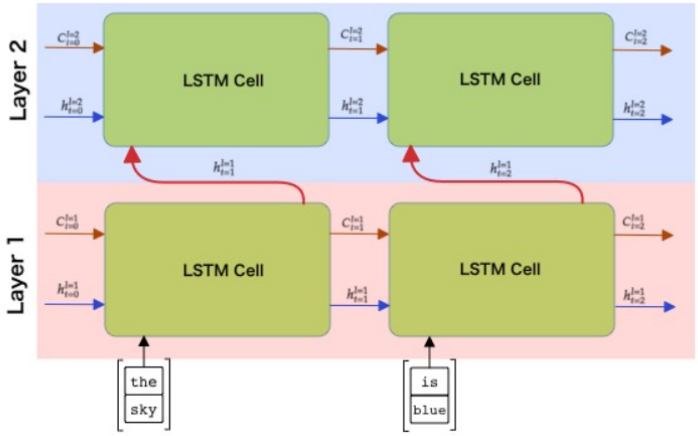


Figure 5.12: LSTM - Multi layer cells

### 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

$$S_{b_1} = \left\{X_i, Y_i\right\}_{i=1}^{i=n_1} \quad , X_i, Y_i \sim f\left(X, Y | b_1\right)$$

$$S_{b_2} = \left\{X_i, Y_i\right\}_{i=1}^{i=n_2} \quad , X_i, Y_i \sim f\left(X, Y | b_2\right)$$

$$S_{b_1} \cup S_{b_2}$$

$$X$$

$$Y$$

$$\left\{b_1, b_2\right\}$$

Trained Model $\mapsto$ Interpetabilty attributes

$$attributes_{\{S_{b_1} \cup S_{b_2}\}}$$

# Bibliography


[Ade+18a] 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.

[Ade+18b] Julius Adebayo et al. *Sanity Checks for Saliency Maps*. 2018. DOI: 10.48550/ARXIV.1810.03292. URL: https://arxiv.org/abs/1810.03292.

[Ade+20] 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.

[Anc+17] 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.

[Bae+10] David Baehrens et al. "How to explain individual classification decisions". In: *The Journal of Machine Learning Research* 11 (2010), pp. 1803–1831.

[CGT09] 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.

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

[Hoo+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.

[HSR21] 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.

[Kow+18] 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.

[LL17] Scott M. Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *CoRR* abs/1705.07874 (2017). arXiv: 1705.07874. URL: http://arxiv.org/abs/1705.07874.

[Mol19] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/. 2019.

[Sam+15] 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.

[SK10] Erik Strumbelj and Igor Kononenko. "An Efficient Explanation of Individual Classifications using Game Theory". In: *J. Mach. Learn. Res.* 11 (2010), pp. 1–18.

[Smi+17] 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.

[STY17] 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.

[SVZ13] 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.

[WRP19] 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.

[ZZ18] 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.