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 and outperform humans on specific tasks ([He+15], [Kow+18], [WRP19], [Mni+13], [Sil+16]).

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 ? 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 on 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'll review recent work for frameworks of evaluating interpretability methods and what evaluation metrics were used. We're also going to review the interpretability method used in the current framework. We'll constrain the scope of this work only to model attribution-based explanations [Zho+21], 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.

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 (feature) for the prediction $y = f_\theta(x)$.

3.1 Interpretability methods frameworks and metrics

3.1.1 Feature importance removal

There has been a line of study to treat attribution-based explanations as feature importance [Sam+15] [Hoo+18] [MSM17]. 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. [MSM17] coined this desired property as *Explanation Selectivity.*

Samec et al. [Sam+15] 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 [Lap+15] to evaluate the layer-wise relevance propagation (LRP) interpretability method "pixel-flipping" was used. Arras et el. [Arr+16] also used LRP evaluation and simply "deleted" word for their model into by setting the corresponding word embedding to zero in order.

Hooker et al. [Hoo+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 they 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

3.1.2 Interpretability as a debug tool

Another line of study is to let the interpretability method to be a debug tool for the model or training process itself [Ade+18b], [Ade+20]. 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. [Ade+18b] 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.

[Ade+18b] 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 [Wan+04]).

3. Pearson correlation of the histogram of gradients (HOGs)

Paragraph about Data contamination and test bugs [Ade+20]

3.1.3 A more theoretical and axiomatic approach

Paragraph about Axioms in IG [STY17]

Paragraph about Monotonicity [NM20]

Paragraph about Sensitivity-n [Anc+17]

Paragraph about Explanation Continuity [MSM17]

3.2 Interpretability methods

For the this section we'll assume we have a model f

3.2.1 Grad

The most straight forward interpretability method [Bae+10] [SVZ13], also know as saliency map

3.2.2 Integrated Gradients

Paragraph about method [STY17]

3.2.3 Local Integrated Gradients

Paragraph about method [Anc+17]

3.2.4 Smooth Grad

Paragraph about method [Smi+17]

3.2.5 VarGrad

Paragraph about method [Ade+18a]

3.2.6 Smooth Grad Square SG-SQ

Paragraph about method [Hoo+18]

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:

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

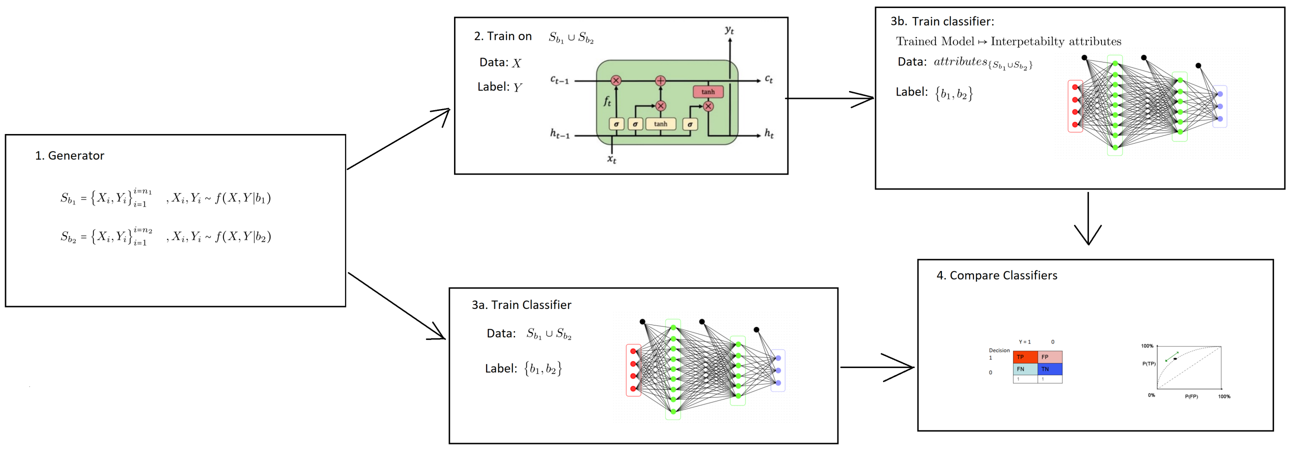

1. Generator
$S_{b_1} = \{X_i, Y_i\}_{i=1}^{i=n_1} \quad , X_i, Y_i \sim f(X, Y|b_1)$
$S_{b_2} = \{X_i, Y_i\}_{i=1}^{i=n_2} \quad , X_i, Y_i \sim f(X, Y|b_2)$
2. Train on $S_{b_1} \cup S_{b_2}$
Data: X
Label: Y
y_t
c_{t-1}
c_t
f_t
tanh
σ
σ
tanh
σ
h_{t-1}
h_t
x_t
3b. Train classifier:
Trained Model $\mapsto$ 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

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

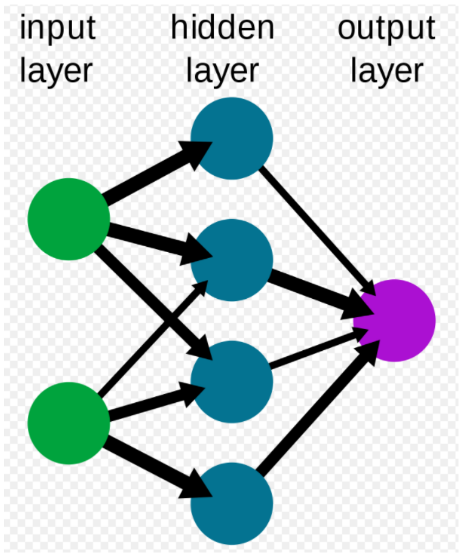


Figure 5.1: Neural Network Architecture

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

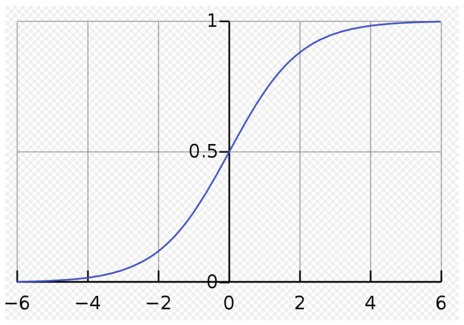


Figure 5.2: Sigmoid function

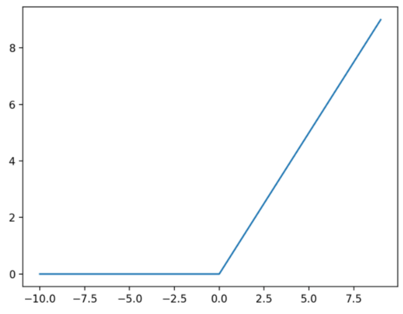


Figure 5.3: ReLU function

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

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

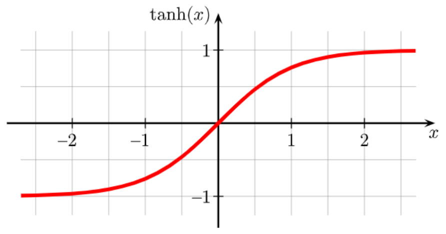


Figure 5.4: Hyperbolic tangent function

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)

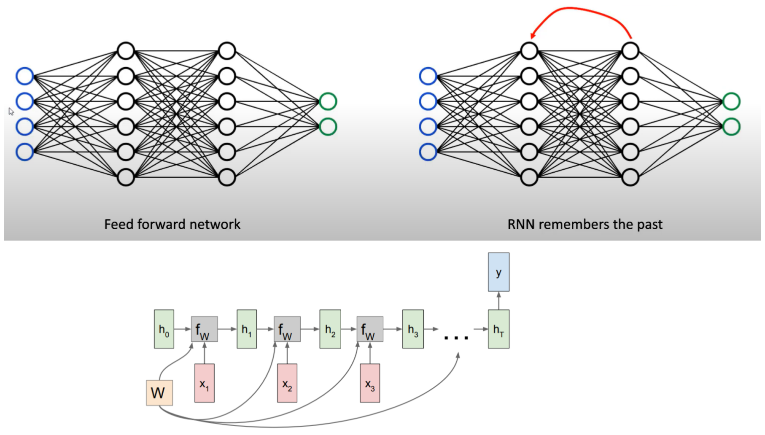


Figure 5.5: RNN

RNN - Simple example

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

RNN - Multi-Layer

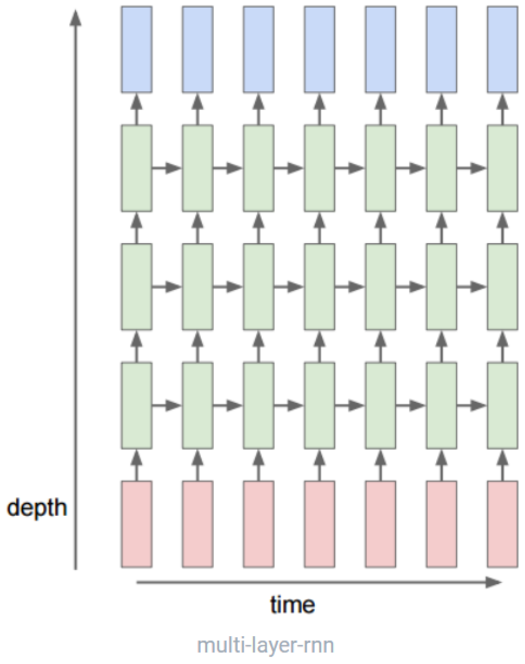


Figure 5.6: RNN - MultiLayer

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

RNN - Training

Back propagation through time:

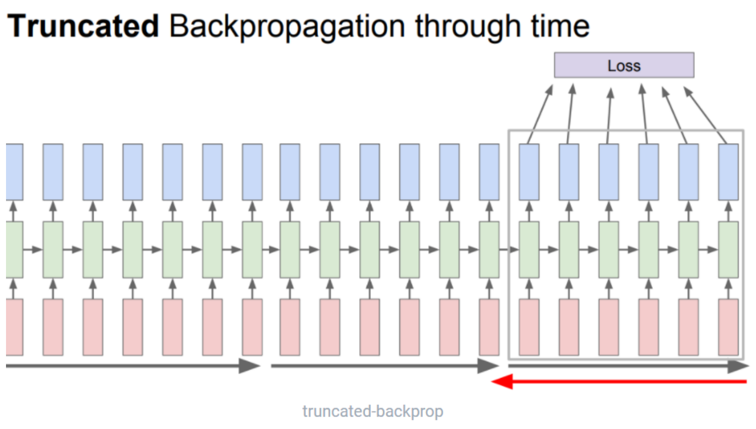
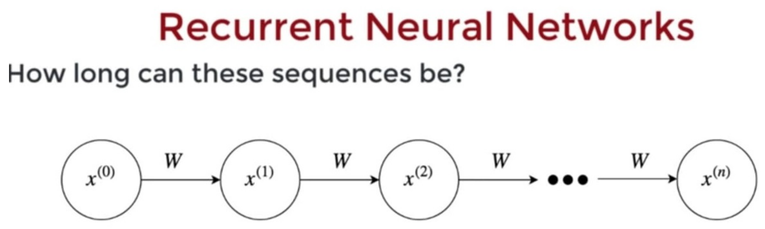


Figure 5.7: RNN - back propagation

RNN - Problems

Small example, scalars and no hidden states:



$$\begin{tikzcd} x^{(0)} \arrow[r, "W"] & x^{(1)} \arrow[r, "W"] & x^{(2)} \arrow[r, "W"] & \bullet\bullet\bullet \arrow[r, "W"] & x^{(n)} \end{tikzcd}$$

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

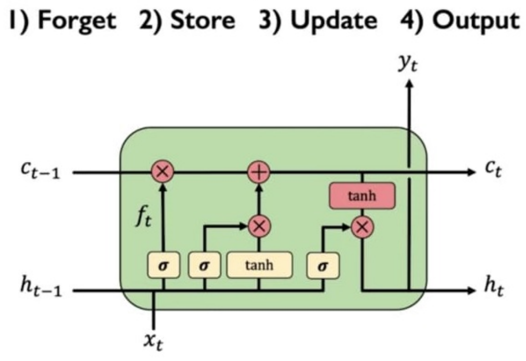


Figure 5.9: LSTM - Scheme

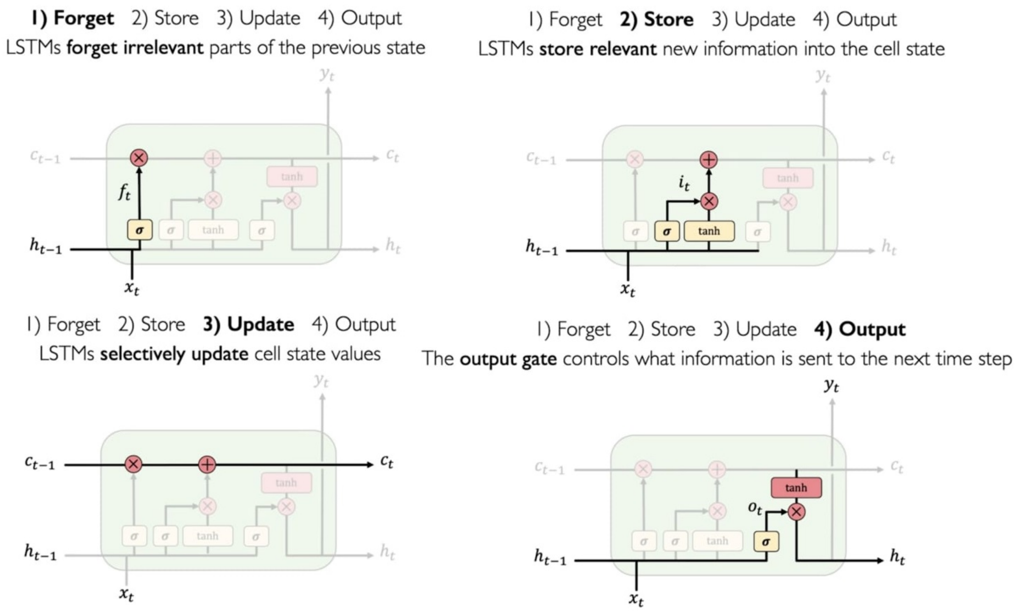

1) Forget 2) Store 3) Update 4) Output
LSTMs forget irrelevant parts of the previous state
LSTMs store relevant new information into the cell state
LSTMs selectively update cell state values
The output gate controls what information is sent to the next time step

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**

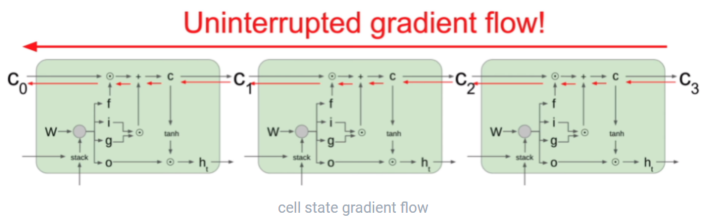


Figure 5.11: LSTM - Gradient flow

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

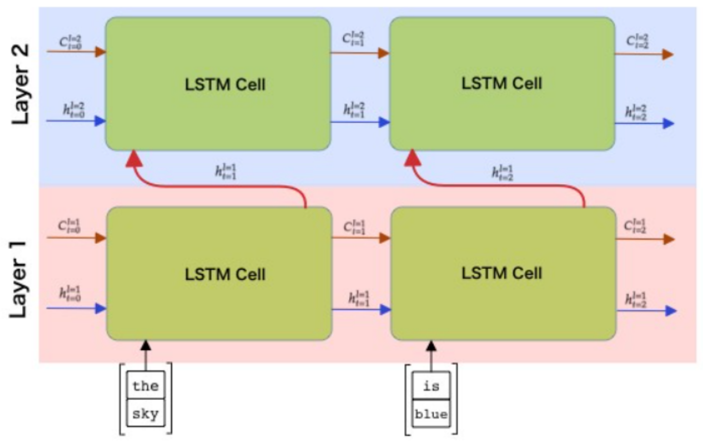


Figure 5.12: LSTM - Multi layer cells

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

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

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

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

X
Y
$\{b_1, b_2\}$

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.

[Arr+16] 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.

[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.

[Lap+15] 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.

[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.

[Mni+13] 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.

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

[MSM17] 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.

[NM20] 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`.

[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`.

[Sil+16] 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`.

[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`.

[Wan+04] 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`.

[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`.

[Zho+21] 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`.

[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`.