BlackboxNLP 2019

August 1, Florence, Italy, @ ACL 2019

Evaluating Recurrent Neural Network Explanations

Leila Arras , Ahmed Osman, Klaus-Robert Müller and Wojciech Samek

Fraunhofer HHI

berlin

Explaining predictions

Blackbox

"it's a bike" but why?

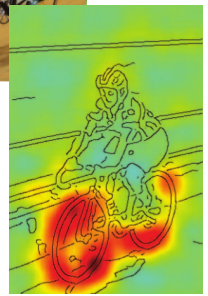
- Decision for right reason or biased?
- EU-GDPR 2016 compliant?

$\boldsymbol{x} \in \mathbb{R}^D$
input

Transparency

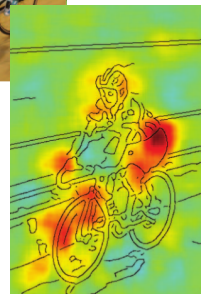
"it's a bike"

$f_c(\boldsymbol{x}) \in \mathbb{R}$
prediction score



$\boldsymbol{R} \in \mathbb{R}^D$
relevance

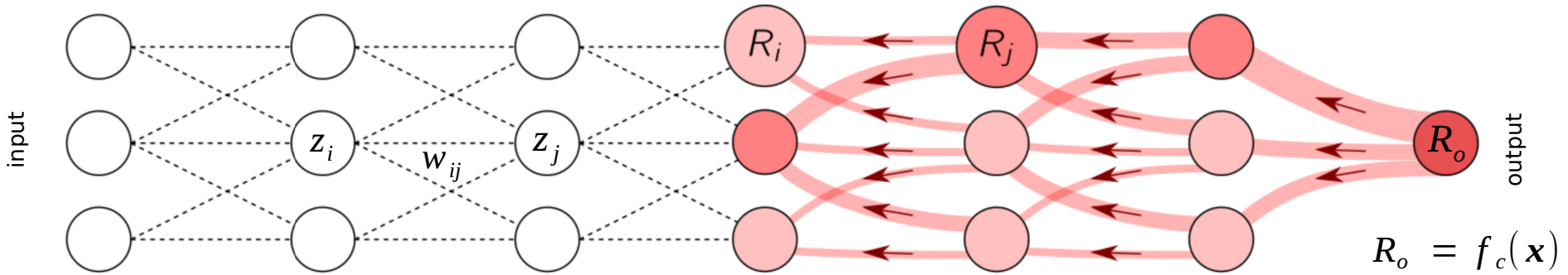
"it's a person"



[Lapuschkin et al. 2016]

Layer-wise Relevance Propagation (LRP)

Idea: Decompose prediction function value $f_c(\boldsymbol{x})$. [Bach et al. 2015]



$$R_o = f_c(\boldsymbol{x})$$

$$z_j = g\Big(\underbrace{\sum_i w_{ij} z_i + b_j}_{x_j}\Big)$$

Forward pass

$$R_i = \sum_j \frac{w_{ij} z_i}{x_j + \epsilon \cdot sign(x_j)} R_j$$

Backward pass

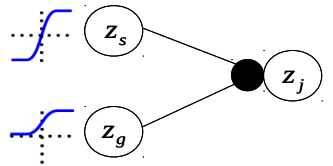
Theoretical justification for ReLU nets based on Taylor expansion [Montavon et al. 2017]

Extending LRP to recurrent networks

New: How to propagate relevance through products?

Forward pass

$$z_j = z_s \cdot z_g$$



signal (signed) "is the information"

gate ϵ (0,1) "controls the flow of information"

	LRP-all	**LRP-prop**	**LRP-abs**	**LRP-half**
signal	R_s → R_j (1)	R_s → R_j	R_s → R_j	$\frac{R_j}{2}$ → R_j (0.5)
gate	0	R_g → R_j	R_g → R_j	$\frac{R_j}{2}$ → R_j (0.5)
	signal−take−all	$R_s = \frac{z_s}{z_s+z_g} R_j$ $R_g = \frac{z_g}{z_s+z_g} R_j$	$R_s = \frac{\lvert z_s \rvert}{\lvert z_s \rvert + \lvert z_g \rvert} R_j$ $R_g = \frac{\lvert z_g \rvert}{\lvert z_s \rvert + \lvert z_g \rvert} R_j$	*equally*
	Arras et al. 2017b	Ding et al. 2017 Arjona-Medina et al. 2018		Arjona-Medina et al. 2018

Other methods for recurrent networks

- Contextual Decomposition (CD) [Murdoch et al. 2018]
- Gradient, Gradient x Input [Li et al. 2016, Denil et al. 2015, Gevrey et al. 2003]
- Occlusion relevance [Li et al. 2017]

see paper for details

now which one to use?

Evaluating Explanations

Look at a few heatmaps?

Predicted class: negative sentiment

---

this movie was actually neither that funny , nor super witty .

i hate the movie though the plot is interesting .

not worth the time

is n't a bad film (misclassified)

it never fails to engage us . (misclassified)

OK but this is not enough

Perturbation experiment

Idea: **Perturb the input** according to word relevance ordering
Track the impact on the prediction [Arras et al. 2016]

akin **pixel-flipping/ region perturbation** in Computer Vision [Bach et al. 2015, Samek et al. 2017] or **ablation**

Step	Input	Predicted sentiment
0	i hate the movie though the plot is interesting .	- -
1	i ____ the movie though the plot is interesting .	+
2	i ____ the movie though the ____ is interesting .	+
3	i ____ the movie ____ the ____ is interesting .	++

word embedding set to zero

Perturbation Results

LSTM sentiment analysis

Word deletion	Grad x Input	LRP-all	LRP-prop	LRP-abs	LRP-half	CD	occlusion
Impact on accuracy	✘	✓	✘	✘	✘	✓	✓

Model: Bidirectional LSTM from [Li et al. 2016]
Task: 5-class sentiment prediction (SST 5) [Socher et al. 2013]

Perturbation Results

LSTM sentiment analysis

only one backward pass

multiple passes

Word deletion	Grad x Input	LRP-all	LRP-prop	LRP-abs	LRP-half	CD	occlusion
Impact on accuracy	✘	✓	✘	✘	✘	✓	✓

Model: Bidirectional LSTM from [Li et al. 2016]
Task: 5-class sentiment prediction (SST 5) [Socher et al. 2013]

Toy Arithmetic Task

We propose a toy setup, with Ground Truth relevance

Input Sequence

$$\begin{bmatrix} 0 & 0 & 0 & \boxed{n_a} & 0 & 0 & 0 & \boxed{n_b} & 0 & 0 & 0 \\ n_1 & \dots & n_{a-1} & 0 & n_{a+1} & \dots & n_{b-1} & 0 & n_{b+1} & \dots & n_T \end{bmatrix}$$

$$\underbrace{\quad}_{x_1} \quad \underbrace{\quad}_{x_2} \qquad x_a \qquad\qquad x_b \qquad\qquad \underbrace{\quad}_{x_T}$$

Task 1 $\boxed{y_{target} = n_a + n_b}$ addition $n_t \in \mathbb{R}$

Model LSTM with one cell

Similar to adding problem [Hochreiter and Schmidhuber 1996]

Toy Arithmetic Task

We propose a toy setup, with Ground Truth relevance

Input Sequence

$$R_{x_a} = n_a \qquad R_{x_b} = n_b$$

$$\begin{bmatrix} 0 & 0 & 0 & n_a & 0 & 0 & 0 & n_b & 0 & 0 & 0 \\ n_1 & \dots & n_{a-1} & 0 & n_{a+1} & \dots & n_{b-1} & 0 & n_{b+1} & \dots & n_T \end{bmatrix}$$

$$\underbrace{x_1}_{n_1} \quad \underbrace{x_2}_{\dots} \quad x_a \quad x_b \quad \underbrace{x_T}_{n_T}$$

Task 1 $y_{target} = n_a + n_b$ addition $n_t \in \mathbb{R}$

Model LSTM with one cell

Toy Arithmetic Task

We propose a toy setup, with Ground Truth relevance

Input Sequence

$$R_{x_a} = n_a \qquad R_{x_b} = -n_b$$

$$\begin{bmatrix} 0 & 0 & 0 & n_a & 0 & 0 & 0 & n_b & 0 & 0 & 0 \\ n_1 & \dots & n_{a-1} & 0 & n_{a+1} & \dots & n_{b-1} & 0 & n_{b+1} & \dots & n_T \end{bmatrix}$$

$$x_1 \quad x_2 \quad x_a \quad x_b \quad x_T$$

Task 2 $y_{target} = n_a - n_b$ subtraction $n_t \in \mathbb{R}^+$

Model LSTM with one cell

Results

	Toy Task Addition	Correlation between relevance and input number $\rho(R_{\boldsymbol{x_a}}, n_a)$ (in %)	$\rho(R_{\boldsymbol{x_b}}, n_b)$ (in %)
✓	Gradient×Input	**99.960** (0.017)	**99.954** (0.019)
✓	Occlusion	**99.990** (0.004)	**99.990** (0.004)
✘	LRP-prop	0.785 (3.619)	10.111 (12.362)
✘	LRP-abs	7.002 (6.224)	12.410 (17.440)
✘	LRP-half	29.035 (9.478)	51.460 (19.939)
✓	LRP-all	**99.995** (0.002)	**99.995** (0.002)
✓	CD	**99.997** (0.002)	**99.997** (0.002)

Results

	Toy Task Addition	$\rho(R_{\boldsymbol{x_a}}, n_a)$ (in %)	$\rho(R_{\boldsymbol{x_b}}, n_b)$ (in %)
✓	Gradient×Input	**99.960** (0.017)	**99.954** (0.019)
✓	Occlusion	**99.990** (0.004)	**99.990** (0.004)
✘	LRP-prop	0.785 (3.619)	10.111 (12.362)
✘	LRP-abs	7.002 (6.224)	12.410 (17.440)
✘	LRP-half	29.035 (9.478)	51.460 (19.939)
✓	LRP-all	**99.995** (0.002)	**99.995** (0.002)
✓	CD	**99.997** (0.002)	**99.997** (0.002)

	Toy Task Subtraction	$\rho(R_{\boldsymbol{x_a}}, n_a)$ (in %)	$\rho(R_{\boldsymbol{x_b}}, n_b)$ (in %)
✓	Gradient×Input	**97.9** (1.6)	**-98.8** (0.6)
✘	Occlusion	99.0 (2.0)	-69.0 (19.1)
✘	LRP-prop	3.1 (4.8)	-8.4 (18.9)
✘	LRP-abs	1.2 (7.6)	-23.0 (11.1)
✘	LRP-half	7.7 (15.3)	-28.9 (6.4)
✓	LRP-all	**98.5** (3.5)	**-99.3** (1.3)
✘	CD	-25.9 (39.1)	-50.0 (29.2)

Results

Why different results?

	Toy Task Addition	$\rho(R_{\boldsymbol{x_a}}, n_a)$ (in %)	$\rho(R_{\boldsymbol{x_b}}, n_b)$ (in %)
✓	Gradient×Input	**99.960** (0.017)	**99.954** (0.019)
✓	Occlusion	**99.990** (0.004)	**99.990** (0.004)
✘	LRP-prop	0.785 (3.619)	10.111 (12.362)
✘	LRP-abs	7.002 (6.224)	12.410 (17.440)
✘	LRP-half	29.035 (9.478)	51.460 (19.939)
✓	LRP-all	**99.995** (0.002)	**99.995** (0.002)
✓	CD	**99.997** (0.002)	**99.997** (0.002)

Addition can be solved by a bag of words

	Toy Task Subtraction	$\rho(R_{\boldsymbol{x_a}}, n_a)$ (in %)	$\rho(R_{\boldsymbol{x_b}}, n_b)$ (in %)
✓	Gradient×Input	**97.9** (1.6)	**-98.8** (0.6)
✘	Occlusion	99.0 (2.0)	-69.0 (19.1)
✘	LRP-prop	3.1 (4.8)	-8.4 (18.9)
✘	LRP-abs	1.2 (7.6)	-23.0 (11.1)
✘	LRP-half	7.7 (15.3)	-28.9 (6.4)
✓	LRP-all	**98.5** (3.5)	**-99.3** (1.3)
✘	CD	-25.9 (39.1)	-50.0 (29.2)

Subtraction is sequential (order matters!)

Summary

LSTM for sentiment analysis & toy tasks

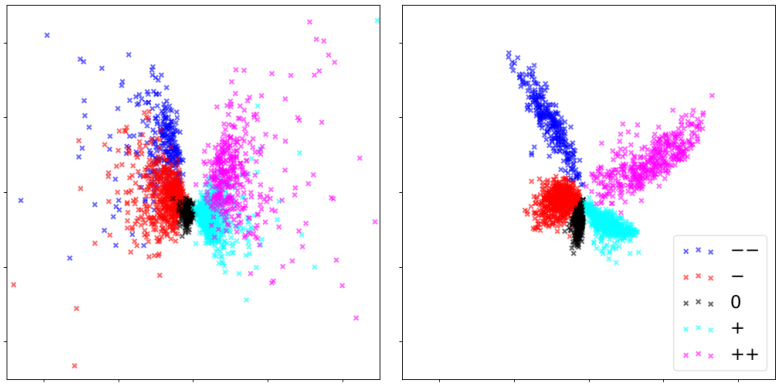
Evaluation	LRP-all	LRP-prop	LRP-abs	LRP-half	CD	Grad x Input	occlusion
Perturbation	✓	✘	✘	✘	✓	✘	✓
Addition	✓	✘	✘	✘	✓	✓	✓
Subtraction	✓	✘	✘	✘	✘	✓	✘
Subject-verb agreement	✓	–	–	–	–	✘	✘

Evaluation by Poerner et al. 2018

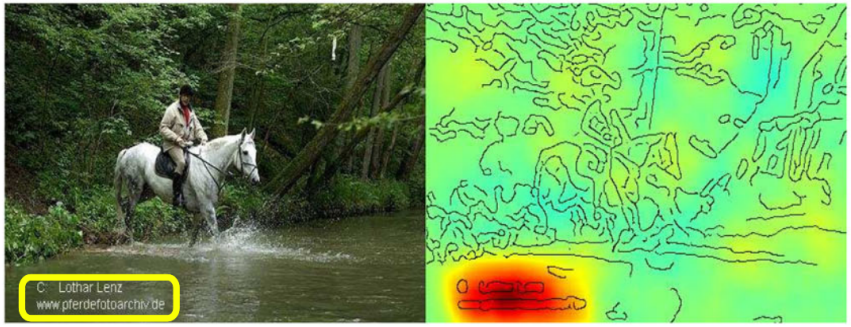
Model: LSTM, GRU, Quasi-RNN
Task: verb number prediction [Linzen et al. 2016]

Applications of relevance

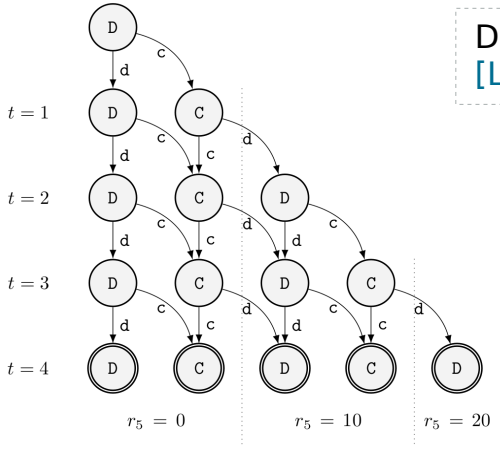
Applications



Sentence-level/Document-level vector representations
[present work, Arras et al. 2017a]



Detect model/dataset bias
[Lapuschkin et al. 2019]



Redistribute rewards in RL
[Arjona-Medina et al. 2018]

Conclusion

- LRP on LSTMs [Bach et al. 2015, Arras et al. 2017b] works well
  → but correct redistribution of relevance through product layer is crucial
- Some methods fail on a simple toy task
  (CD [Murdoch et al. 2018], Occlusion [Li et al. 2017])
- Still, we need more evaluation tasks for relevance!
- Applications could also be extended.

Thanks for your attention!

More information on LRP (code, tutorials, …):
www.heatmapping.org

Reference implementation of LRP for LSTMs:
github.com/ArrasL/LRP_for_LSTM

Tutorial paper:
Methods for interpreting and understanding deep neural networks
Grégoire Montavon, Wojciech Samek and Klaus-Robert Müller
Digital Signal Processing 73 (2018) 1-15

Upcoming book:
Chapter on “Explaining and Interpreting LSTMs“
Leila Arras*, Jose Arjona-Medina*, Michael Widrich, Grégoire Montavon, Michael Gillhofer, Klaus-Robert Müller, Sepp Hochreiter and Wojciech Samek (*equal contribution)