Evaluating Recurrent Neural Network Explanations

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

```
x ∈ IR^D
input

R ∈ IR^D
relevance

f_c(x) ∈ IR
prediction score

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

"it's a bike" but why?
"it's a bike"
"it's a person"
```

[Lapuschkin et al. 2016]
Layer-wise Relevance Propagation (LRP)

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

Forward pass

\[ z_j = g\left(\sum_i w_{ij} z_i + b_j\right) \]

Backward pass

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

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?

\[ Z_j = Z_s \cdot Z_g \]

**Forward pass**

**signal** (signed) “is the information”

**gate** $\epsilon (0,1)$ “controls the flow of information”

**LRP-all**

- **signal**
  - Arras et al. 2017b

**LRP-prop**

- **signal**
  - $R_s = \frac{z_s}{z_s + z_g} R_j$
  - Ding et al. 2017

**LRP-abs**

- **signal**
  - $R_s = \frac{|z_s|}{|z_s| + |z_g|} R_j$
  - Arjona-Medina et al. 2017b

**LRP-half**

- **signal**
  - $R_s = \frac{R_j}{2}$
  - Arjona-Medina et al. 2018

**gate**

- **LRP-all**
  - $R_g = z_g$

- **LRP-prop**
  - $R_g = \frac{z_g}{z_s + z_g} R_j$

- **LRP-abs**
  - $R_g = \frac{|z_g|}{|z_s| + |z_g|} R_j$

- **LRP-half**
  - $R_g = \frac{R_j}{2}$

**signal - take - all**

**equally**
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**

<table>
<thead>
<tr>
<th>Step</th>
<th>Input</th>
<th>Predicted sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>i hate the movie though the <strong>plot is interesting</strong> .</td>
<td>- -</td>
</tr>
<tr>
<td>1</td>
<td>i ___ the movie though the <strong>plot is interesting</strong> .</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>i ___ the movie though the ___ is interesting .</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>i ___ the movie ___ the ___ is interesting .</td>
<td>++</td>
</tr>
</tbody>
</table>

word embedding set to zero
Perturbation Results

LSTM sentiment analysis

<table>
<thead>
<tr>
<th>Word deletion</th>
<th>Grad x Input</th>
<th>LRP-all</th>
<th>LRP-prop</th>
<th>LRP-abs</th>
<th>LRP-half</th>
<th>CD</th>
<th>occlusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impact on accuracy</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

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

LSTM sentiment analysis

<table>
<thead>
<tr>
<th>Word deletion</th>
<th>Grad x Input</th>
<th>LRP-all</th>
<th>LRP-prop</th>
<th>LRP-abs</th>
<th>LRP-half</th>
<th>CD</th>
<th>occlusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impact on accuracy</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Model: Bidirectional LSTM from [Li et al. 2016]
Task: 5-class sentiment prediction (SST 5) [Socher et al. 2013]
We propose a toy setup, with Ground Truth relevance

\[
\begin{bmatrix}
0 & 0 & 0 & n_a & 0 & 0 & 0 & n_b & 0 & 0 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
n_1 & \ldots & n_{a-1} & 0 & n_{a+1} & \ldots & n_{b-1} & 0 & n_{b+1} & \ldots & n_T
\end{bmatrix}
\]

\[
x_1 \quad x_2 \quad x_a \quad x_b \quad x_T
\]

Task 1

\[y_{\text{target}} = n_a + n_b\]

addition \hspace{1cm} n_t \in \mathbb{R}

Model

LSTM with one cell

Similar to adding problem [Hochreiter and Schmidhuber 1996]
We propose a toy setup, with Ground Truth relevance.

### Input Sequence

$$
\begin{bmatrix}
0 & 0 & 0 & n_a & 0 & 0 & 0 & n_b & 0 & 0 & 0 & 0 \\
n_1 & \ldots & n_{a-1} & 0 & n_{a+1} & \ldots & n_{b-1} & 0 & n_{b+1} & \ldots & n_T \\
x_1 & x_2 & x_a & x_b & x_{a+1} & x_{b+1} & x_{T-1} & x_T
\end{bmatrix}
$$

### 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

\[
\begin{bmatrix}
0 & 0 & 0 & n_a & 0 & 0 & 0 & n_b & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_1 & x_2 & x_a & x_b & x_{b+1} & \ldots & x_{T-1} & x_T \\
\end{bmatrix}
\]

Task 2

\[
y_{\text{target}} = n_a - n_b
\]

subtraction

\(n_t \in \mathbb{R}^+\)

Model

LSTM with one cell
## Results

<table>
<thead>
<tr>
<th>Toy Task Addition</th>
<th>$\rho(R_{x\alpha}, n_\alpha)$ (in %)</th>
<th>$\rho(R_{x\beta}, n_\beta)$ (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient $\times$ Input</td>
<td>99.960 (0.017)</td>
<td>99.954 (0.019)</td>
</tr>
<tr>
<td>Occlusion</td>
<td>99.990 (0.004)</td>
<td>99.990 (0.004)</td>
</tr>
<tr>
<td>LRP-prop</td>
<td>0.785 (3.619)</td>
<td>10.111 (12.362)</td>
</tr>
<tr>
<td>LRP-abs</td>
<td>7.002 (6.224)</td>
<td>12.410 (17.440)</td>
</tr>
<tr>
<td>LRP-half</td>
<td>29.035 (9.478)</td>
<td>51.460 (19.939)</td>
</tr>
<tr>
<td>LRP-all</td>
<td>99.995 (0.002)</td>
<td>99.995 (0.002)</td>
</tr>
<tr>
<td>CD</td>
<td>99.997 (0.002)</td>
<td>99.997 (0.002)</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Toy Task Addition</th>
<th>$\rho(R_{x\alpha}, n_{\alpha})$ (in %)</th>
<th>$\rho(R_{x\beta}, n_{\beta})$ (in %)</th>
<th>Toy Task Subtraction</th>
<th>$\rho(R_{x\alpha}, n_{\alpha})$ (in %)</th>
<th>$\rho(R_{x\beta}, n_{\beta})$ (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Gradient $\times$ Input</td>
<td>99.960 (0.017)</td>
<td>99.954 (0.019)</td>
<td>✓ Gradient $\times$ Input</td>
<td>97.9 (1.6)</td>
<td>-98.8 (0.6)</td>
</tr>
<tr>
<td>✓ Occlusion</td>
<td>99.990 (0.004)</td>
<td>99.990 (0.004)</td>
<td>✓ Occlusion</td>
<td>99.0 (2.0)</td>
<td>-69.0 (19.1)</td>
</tr>
<tr>
<td>✗ LRP-prop</td>
<td>0.785 (3.619)</td>
<td>10.111 (12.362)</td>
<td>✗ LRP-prop</td>
<td>3.1 (4.8)</td>
<td>-8.4 (18.9)</td>
</tr>
<tr>
<td>✗ LRP-abs</td>
<td>7.002 (6.224)</td>
<td>12.410 (17.440)</td>
<td>✗ LRP-abs</td>
<td>1.2 (7.6)</td>
<td>-23.0 (11.1)</td>
</tr>
<tr>
<td>✗ LRP-half</td>
<td>29.035 (9.478)</td>
<td>51.460 (19.939)</td>
<td>✗ LRP-half</td>
<td>7.7 (15.3)</td>
<td>-28.9 (6.4)</td>
</tr>
<tr>
<td>✓ LRP-all</td>
<td>99.995 (0.002)</td>
<td>99.995 (0.002)</td>
<td>✓ LRP-all</td>
<td>98.5 (3.5)</td>
<td>-99.3 (1.3)</td>
</tr>
<tr>
<td>✓ CD</td>
<td>99.997 (0.002)</td>
<td>99.997 (0.002)</td>
<td>✓ CD</td>
<td>-25.9 (39.1)</td>
<td>-50.0 (29.2)</td>
</tr>
</tbody>
</table>
### Results

**Why different results?**

<table>
<thead>
<tr>
<th>Toy Task Addition</th>
<th>$\rho(R_{\alpha\alpha}, n_{\alpha})$ (in %)</th>
<th>$\rho(R_{\alpha\beta}, n_{\beta})$ (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Gradient × Input</td>
<td><strong>99.960</strong> (0.017)</td>
<td><strong>99.954</strong> (0.019)</td>
</tr>
<tr>
<td>✓ Occlusion</td>
<td><strong>99.990</strong> (0.004)</td>
<td><strong>99.990</strong> (0.004)</td>
</tr>
<tr>
<td>✗ LRP-prop</td>
<td>0.785 (3.619)</td>
<td>10.111 (12.362)</td>
</tr>
<tr>
<td>✗ LRP-abs</td>
<td>7.002 (6.224)</td>
<td>12.410 (17.440)</td>
</tr>
<tr>
<td>✗ LRP-half</td>
<td>29.035 (9.478)</td>
<td>51.460 (19.939)</td>
</tr>
<tr>
<td>✓ LRP-all</td>
<td><strong>99.995</strong> (0.002)</td>
<td><strong>99.995</strong> (0.002)</td>
</tr>
<tr>
<td>✓ CD</td>
<td><strong>99.997</strong> (0.002)</td>
<td><strong>99.997</strong> (0.002)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Toy Task Subtraction</th>
<th>$\rho(R_{\alpha\alpha}, n_{\alpha})$ (in %)</th>
<th>$\rho(R_{\alpha\beta}, n_{\beta})$ (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Gradient × Input</td>
<td><strong>97.9</strong> (1.6)</td>
<td>-98.8 (0.6)</td>
</tr>
<tr>
<td>✗ Occlusion</td>
<td>99.0 (2.0)</td>
<td>-69.0 (19.1)</td>
</tr>
<tr>
<td>✗ LRP-prop</td>
<td>3.1 (4.8)</td>
<td>-8.4 (18.9)</td>
</tr>
<tr>
<td>✗ LRP-abs</td>
<td>1.2 (7.6)</td>
<td>-23.0 (11.1)</td>
</tr>
<tr>
<td>✗ LRP-half</td>
<td>7.7 (15.3)</td>
<td>-28.9 (6.4)</td>
</tr>
<tr>
<td>✓ LRP-all</td>
<td><strong>98.5</strong> (3.5)</td>
<td><strong>99.3</strong> (1.3)</td>
</tr>
<tr>
<td>✗ CD</td>
<td>-25.9 (39.1)</td>
<td>-50.0 (29.2)</td>
</tr>
</tbody>
</table>

**Addition can be solved by a bag of words**

**Subtraction is sequential (order matters!)**
## Summary

LSTM for sentiment analysis & toy tasks

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>LRP-all</th>
<th>LRP-prop</th>
<th>LRP-abs</th>
<th>LRP-half</th>
<th>CD</th>
<th>Grad x Input</th>
<th>occlusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perturbation</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Addition</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Subtraction</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Subject-verb agreement</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

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)