Designing RNNs for Explainability

Pattarawat Chormai
Supervised by Dr. Grégoire Montavon and Prof. Klaus-Robert Müller
Department of Machine Learning
This work is a part of master’s thesis [4]. Contact email: pat.chormai@gmail.com

Introduction

RNNs have become very popular in various sequence-processing applications. However, it is still unclear how their decisions/outputs are produced, raising concerns among stakeholders.

Because RNNs need to summarize information across timesteps, we hypothesize that well-structured RNNs would have an advantage of being more explainable although they perform equivalently in terms of objective function. In particular, we study and answer the following questions:

- How does the structure of RNNs affect their explainability?
- Is LSTM more explainable than standard RNNs?
- How does the structure of RNNs affect their explainability?

Explanation Methods

Neural networks or RNNs can be viewed as $\mathbf{x} \mapsto f(\mathbf{x})$. Explanation methods aim to find relevance scores $R_k(\mathbf{x})$ quantifying the importance of every component in $\mathbf{x}$ to $f(\mathbf{x})$.

- Sensitivity Analysis (SA) [1]: $R_k(\mathbf{x}) = \left( \frac{\partial f(\mathbf{x})}{\partial x_k} \right)^2$

- Guided Backprop (GB) [2] is proposed for ReLU-type architectures. It is based on computing the derivatives, but local gradients are backpropagated only when incoming activations and the signal are not positive.

- Deep Taylor Decomposition (DTD) [3] is derived specifically for explaining neural networks with ReLU activations. It redistributes $f(\mathbf{x})$ to $\mathbf{x}$ using certain propagation rules derived from the Taylor expansion. Given $\mathbf{j}$ and $\mathbf{k}$ are neurons in two consecutive hidden layers, DTD distributes relevance scores as follows:

$$R_j(\mathbf{x}) = \sum_k a_j w_{jk}^+ \prod_{l \neq j} R_l(\mathbf{x})$$

where $a_j$ is neuron $j$’s activation and $w_{jk}^+ = \max(0, w_{jk})$.

Experimental Setup

We construct an artificial classification problem using MNIST and FashionMNIST. The classification is to determine the majority sample in the sequence. We name this problem MNIST-MAJ.

Relevance heatmap

What pixels make the RNN think the input corresponds to ‘9’?

Fig. 1: MNIST-MAJ classification problem and how explanation methods are applied.

Architectures

We consider five RNN architectures including Shallow, Deep, ConvDeep and a modified LSTM (R-LSTM) where tanh activations are replaced by ReLU. We also experiment R-LSTM with convolutional layers (ConvR-LSTM).

Results

We train models to reach accuracy approximately 98% for MNIST and 85% for FashionMNIST using sequence length 12 ($\mathbf{x}_t \in \mathbb{R}^{28 \times 7}$).

Fig. 2: Shallow, Deep, ConvDeep, R-LSTM architectures with the number of neurons in each layers depicted.

Fig. 3: Relevance heatmaps of MNIST-MAJ from different models and explanation methods.

Discussion

- Our results show that deeper RNN and LSTM-type architectures have more explainable predictions even though their accuracy is equivalent.
- The results also suggest that DTD is more sensitive to the architecture of RNNs than SA and GB.

Acknowledgement

I would like to thank Prof. Dr. Klaus-Robert Müller and Dr. Grégoire Montavon for guidance and support throughout the study.

References