State Gradients for RNN Memory Analysis

Lyan Verwimp, Hugo Van hamme, Vincent Renkens, Patrick Wambacq
ESAT – PSI, KU Leuven
Kasteelpark Arenberg 10
3001 Heverlee, Belgium
{firstname}.{lastname}@kuleuven.be

Abstract

We present a framework for analyzing what the state in RNNs remembers from its input embeddings. We compute the gradients of the states with respect to the input embeddings and decompose the gradient matrix with Singular Value Decomposition to analyze which directions in the embedding space are best transferred to the hidden state space, characterized by the largest singular values. We apply our approach to LSTM language models and investigate to what extent and for how long certain classes of words are remembered on average for a certain corpus. Additionally, the extent to which a specific property or relationship is remembered by the RNN can be tracked by comparing a vector characterizing that property with the direction(s) in embedding space that are best preserved in hidden state space.

1 Introduction

Recurrent neural networks (RNNs) are the current state of the art in many speech and language technology applications, but they are often called 'black-box' models since it is hard for humans to interpret what exactly the network has learned. We present a framework to investigate what the states of RNNs remember from their input and for how long. We apply our approach to the current state of the art in language modeling, long short-term memory (Hochreiter and Schmidhuber, 1997) (LSTM) LMs (Sundermeyer et al., 2012), but it can be applied to other types of RNNs too and to other models with continuous word representations as input.

2 Average memory of the RNN

Our framework is inspired by backpropagation, but instead of computing the gradient of the loss, we compute the gradient of the state with respect to the input embedding, the 'state gradient', to capture the influence of the input on the state. To examine how long input words are remembered by the RNN, we calculate the gradient with a certain delay – with respect to the input word embedding a few time steps earlier. The gradient matrix $\bar{\mathbf{G}}_\tau$ (averaged over all time steps), where τ is a certain delay, is decomposed with Singular Value Decomposition (SVD):

$$\bar{\mathbf{G}}_\tau = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T = \sigma_1\,\mathbf{u}_1\,\mathbf{v}_1^T + \sigma_2\,\mathbf{u}_2\,\mathbf{v}_2^T + \ldots \quad (1)$$

We can interpret $\mathbf{V}$ as directions in the embedding space, $\mathbf{\Sigma}$ as the extent to which the directions in the embedding space can be found in the hidden state space and $\mathbf{U}$ as corresponding directions in the hidden state space. Hence, the directions with the largest singular values (SVs) (lowest index) are directions in embedding space that are best remembered by the RNN.

In order to investigate how well the RNN remembers on a corpus level, we can track the largest SV or the sum of all SVs with respect to the delay τ. For an LM trained on Penn Treebank, we observe an exponential decay of the SVs with respect to the delay: much of the information that is present in the cell state about a specific word is quickly forgotten. However, on average, some information is still remembered even after processing more than 20 words. The ratio of the largest SV with respect to the sum of all SVs becomes larger as the delay increases, indicating that the memory becomes more selective.

We can also compare the SVs based on gradient matrices averaged over specific classes of words or individual words. We observe for example that pronouns have a larger effect on the cell state than other parts-of-speech for a delay of 0, which makes sense because they determine which verb conjugation should follow.

Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 344–346
Brussels, Belgium, November 1, 2018. ©2018 Association for Computational Linguistics

3 Tracking a specific property

We can also track whether a specific relationship encoded in the input embedding is remembered by the RNN. It has been shown that relationships between word embeddings can be characterized as vector offsets (Mikolov et al., 2013). We compare a vector characterizing a specific property to the directions in the embedding space that are best remembered (the directions in $\mathbf{V}^T$ corresponding to the largest SVs), to see if and how well the property is remembered in the hidden state.

Firstly, we define a specific property as the difference between the averaged embeddings for the classes separated by that property:

$$\mathbf{d}_{a-b} = \bar{\mathbf{e}}_a - \bar{\mathbf{e}}_b \qquad (2)$$

where $\bar{\mathbf{e}}_a$ and $\bar{\mathbf{e}}_b$ are the result of averaging all embeddings of words belonging to classes a and b respectively. In order to check whether this definition makes sense for a specific property, we first test whether the embeddings of the two classes are linearly separable by training a linear classifier.

We propose two methods to investigate the extent to which a property is remembered. Firstly, we can compare $\mathbf{d}$ with $\mathcal{H}_n$, which is the subspace of the embedding space spanned by the directions that are best remembered, the n largest right-singular vectors. To be able to do this, we make the orthogonal projection of $\mathbf{d}$ on $\mathcal{H}_n$:

$$\mathbf{y} = \mathrm{proj}_{\mathcal{H}_n}\ \mathbf{d} = \mathbf{V}_n\ \mathbf{V}_n^T\ \mathbf{d} \qquad (3)$$

where $\mathbf{V}_n$ is the matrix containing the n first columns of $\mathbf{V}$. Assuming $\mathbf{d}$ is normalized to unit length, we can calculate the cosine similarity between $\mathbf{y}$ and $\mathbf{d}$ as follows:

$$\cos(\mathbf{d}, \mathbf{y}) = \frac{\mathbf{d}^T\ \mathbf{V}_n\ \mathbf{V}_n^T\ \mathbf{d}}{\|\mathbf{d}\|\ \ \|\mathbf{V}_n\ \mathbf{V}_n^T\ \mathbf{d}\|} = \left\|\mathbf{V}_n^T\ \mathbf{d}\right\| \qquad (4)$$

The cosine similarity between $\mathbf{d}$ and $\mathcal{H}_n$ is a measure of how close $\mathbf{d}$ is to the top n directions that are best remembered in the RNN state.

A second option is comparing $\mathbf{d}$ with the direction in embedding space that is *best* remembered. To do this, we multiply $\mathbf{d}$ with the average gradient matrix:

$$r = \left\|\bar{\mathbf{G}}_\tau \times \mathbf{d}\right\| \qquad (5)$$

If $\mathbf{d}$ would be the embedding direction that is best remembered in the state, then it would be equal to $\mathbf{v}_1$ and r would be equal to σ_1. Hence, in order to get a relative measure of how well the difference between two classes is remembered, we compare r with σ_1 and obtain a 'extent to which the property is remembered, relative to the property that is best remembered', or the 'relative memory' m:

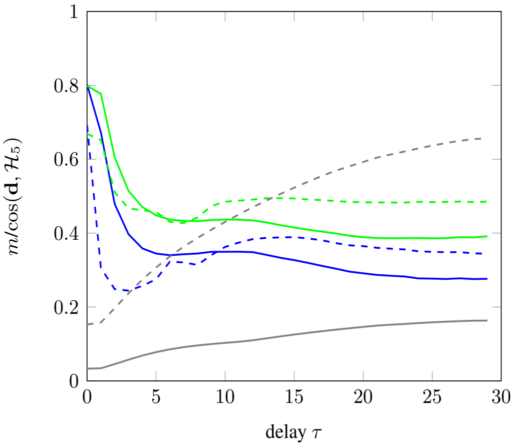

Figure 1: m (full lines) and $\cos(\mathbf{d}, \mathcal{H}_5)$ (dotted lines) for sg-pl (blue) and common-proper (green) nouns with respect to the delay for a PTB LM. Gray lines: $\sigma_1\ /\sum\sigma$ (full) and $\sum_{n=1}^{5}\sigma_n\ /\sum\sigma$ (dotted).

$$m = \frac{r}{\sigma_1} \qquad (6)$$

In Figure 1, we plot m and $\cos(\mathbf{d}, \mathcal{H}_5)$ for the properties singular-plural (sg-pl) noun and common-proper (cm-pr) noun. Prior experiments with a linear classifier showed that these properties can be characterized as a difference vector. According to both measures the sg-pl distinction is slightly better remembered for a delay of 0, while for the other delays the cm-pr distinction is better remembered. In all plots, there is a sharp decrease after a delay of 1 or 2, indicating that the properties seem mostly important on the short term. We also plot the ratio of σ_1 and the sum of the 5 largest SVs with respect to the sum of all SVs (gray lines). Notice that if τ increases, the ratio increases too, which confirms our observation in section 2 that the memory becomes more selective over time.

4 Conclusion

We analyze the memory of an RNN by computing the gradients of its state with respect to its input. The state gradient matrix is decomposed with SVD, and the resulting singular values and directions with the highest singular values are inspected to investigate for how long and how well the RNN remembers its input.

References

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 746–751.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM Neural Networks for Language Modeling. In *INTERSPEECH*, pages 1724–1734.