Predictive Property of Hidden Representations in Recurrent Neural Network Language Models

Sangwoong Yoon
Interdisciplinary Program in Neuroscience
Seoul National University
Seoul, 151-744
swyoon@bi.snu.ac.kr

Sang-Woo Lee
School of Computer Sci. and Eng.
Seoul National University
Seoul, 151-744
slee@bi.snu.ac.kr

Byoung-Tak Zhang
School of Computer Sci. and Eng.
Seoul National University
Seoul, 151-744
btzhang@bi.snu.ac.kr

Abstract

The hidden representation of a recurrent neural network language model (RNNLM) is regarded as a summary of the past input sequence. In this study, we propose that the hidden representation also consists of the expectation about upcoming inputs. A RNNLM is originally trained to predict the next word or character, but we experimentally discover, even for an unmodified RNNLM, the farther sequences can also be predicted given the activation of the hidden neurons. This property makes the hidden activation a summary of the local context covering both the past and the near future, which may benefit some language processing tasks which did not previously take advantage of language models. Dimensionality reduction approach is also briefly considered to facilitate the practical application of the predictive property.

1 Introduction

A Recurrent Neural Network (RNN) is a dynamic neural network with a transient memory, and is naturally suitable for language modeling, which requires memory of the input history. In the past, training a RNN has been difficult due to the long-term dependency [1], and sensitivity to the hyperparameters. However, as this problem pacified with the discoveries of learning heuristics [2] and novel optimization techniques [3], remarkable performances of the RNN language models (RNNLMs) followed, both on the word [4] and character-level [5] language modeling.

The success of RNNs in language modeling indicates that the regularity of language sequences is captured to a non-trivial extent by the model. RNNLM has aided many natural language processing applications such as automatic speech recognition [4] by offering a more accurate probability of a language sequence. Nevertheless, how to incorporate the learned regularity into other language processing systems, which do not explicitly use the probability of a sequence, remains unclear. Many of the sequence labeling problems fall into this case. For example, when segmenting code blocks from plain text [6] or extracting opinions [7], the role of probability of a word is vague.

Analyzing the learned representation of a model provides insights about how the model works. In the case of deep feedforward networks, especially for vision-related tasks, a weight vector or a kernel is often visualized to check what aspect of data the model captures. Moreover, there have been studies investigating a manifold of the representation space [8, 9]. For RNNs, however, few

powerful approaches to examine the representation have been proposed, so our understanding of the nature of the representation is still fragmentary.

Accordingly, in this study, we analyze a RNN and experimentally show that there is a correlation between the hidden representation and unseen future sequences, which we termed as a predictive context. Experiments also show that the predictive context arises from capturing underlying regularity of data. We also suggest the dimensionality reduction approach to make the application of hidden representation more practical.

2 Recurrent Neural Network Language Models

2.1 Recurrent Neural Networks

A recurrent neural network is an extension of a feedforward neural network, with the natural capability of modeling temporal data. It consists of input, hidden, and output neurons like a typical multilayer perceptron, but its hidden neurons possess connections from their past activations. At each time step t, the model takes the sequence value at t as an input, and temporal dependency is learned during the training. Therefore no manual adjustment of a time window is required. Formally,

$$h(t) = \sigma(W_{in}x(t) + W_r h(t-1) + b_1)$$
$$o(t) = \rho(W_{out}h(t) + b_2)$$

Where x, h, and o are input, hidden, and output vectors, and W_{in}, W_r, and W_{out} are weight matrices. b_1 and b_2 are biases. σ and ρ denote nonlinearity functions. Usually, a sigmoid or hyperbolic tangent is used for the former, and a softmax function is used for the latter. In RNNLM, the output neurons represent the probability of next word or character. Other interpretation is also possible for tasks other than language modeling, a RNN encoder-decoder approach being an interesting example of such cases [10].

2.2 Interpreting hidden representations

As the hidden activation vector of a RNN is completely determined by its input history, it is natural to interpret it as a summary of the history. For a RNNLM, the training is to embedding the history into the hidden representation space in a linearly separable manner with respect to the next word or character, because of the logistic regression layer between the hidden and output neurons.

On the other hand, a hidden-to-hidden connection makes neighboring hidden representation correlated to each other. During the training, the correlation is tuned to reflect the underlying patterns of data. It can be deduced that a hidden representation at a certain time step not only reflects the former hidden nodes but also has some regularity with the upcoming. This is how the predictive context emerges.

A bag of surrounding tokens (words or characters), while also has information from both the past and the future, is not equivalent to the hidden representation of RNNLM. First, generalization occurs in the RNN, so the vector might capture more abstract semantic or syntactic regularity. Moreover, language sequences often have non-linear time dependency, which varies according to a given context. Closing a parenthesis is an example of such cases. A bag of fixed number of tokens can not handle the complicated time dependency, while RNNs are capable of handling it.

As considering a hidden representation is, therefore, considering a context at a given point, we suspect the hidden representation of a RNNLM can be a useful feature for other language processing systems. Indeed, the interesting experiment [6] incorporated the hidden vector of RNNLM as an additional feature for a text segmentation system. Although the performance increase was reported, the explanation of how a RNNLM helped the system in lacked. Our analysis on the representation provided one possible explanation.

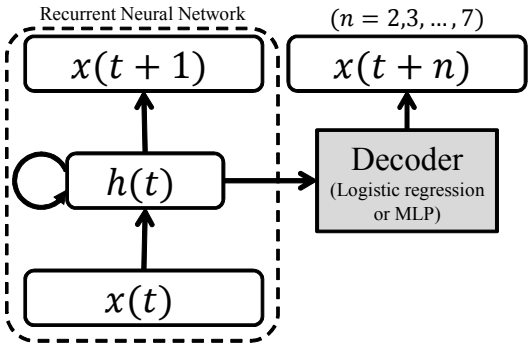

Figure 1: The experimental setting to find the correlation between unseen future characters and the hidden representation of a RNNLM.

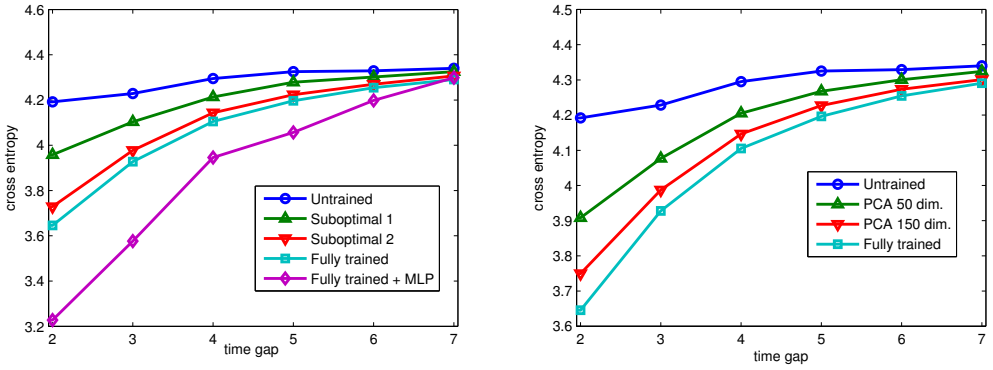

Figure 2: (Left) The hidden vectors of RNNLMs are mapped to unseen future characters with a specific time gap. Each line denotes RNNLMs with different levels of performance. Their test perlexities for language modeling are 43.6 (Untrained), 13.0 (Suboptimal 1), 4.87 (Suboptimal 2), and 2.93 (Fully trained). (Right) Principal component analysis is applied to the hidden representation of a fully trained RNNLM, and the reduced representations are mapped to future characters.

3 Predictive Information in Hidden Representations

3.1 Extracting predictive information

To demonstrate that predictive context information is embedded in the hidden vector of RNNLM, we tried to extract future sequence information with simple classification functions (which will be termed as decoders in this paper) such as logistic regressions or multilayer perceptrons. The decoder maps the hidden vector at time t to the token at $t + n$, where n is a time gap, and its classification cross entropy is taken as a correlation measure between the hidden representation and the future sequence token.

A character-level RNNLM with 300 hidden neurons is constructed using RNNLM Toolkit [4]. After the model is trained on Penntree bank corpus, it is applied again the whole data to collect the hidden activation at each point. Subsequently, a decoder is trained to predict a character at n time step later from the given hidden vector. The separation among train, valid and test set is also maintained while the decoding experiment, so no information from test set affects the training of the decoder.

Given a trained RNNLM, six decoders, each one of which accounts for a specific time gap n ranging from 2 to 7, are trained to show the effect of the gap on the predictive context. As we expected, the hidden vector of RNNLM and future character is correlated. The correlation degrades as the distance become longer. In order to make sure that this effect is caused by the ability of RNN to capture the temporal pattern, we construct suboptimal RNNLMs by restricting the training step, and train

decoders for them. The average cross entropy for the future character diminishes as the performance of a RNN increases, indicating the correlation is caused from the training of a RNN to a data. The whole result and the details on suboptimal models are shown in figure 2 (left).

Considering the fact that logistic regression separates linearly separable data, we can the more a RNNLM is trained, the more linearly separable the hidden representations are with respect to the future character. It is also interesting that even an untrained RNNLM shows a mild cross entropy drop in experiments with smaller time gaps. Since the recurrent weight matrix is randomly initialized, this shows the natural capability of recurrent neurons capturing temporal dependency. Echo-state networks [11], a variant of RNN, are known to exploit such a characteristic.

3.2 Dimensionality reduction approach

The performance of RNN usually increases with the number of hidden units, but it is not practical to use as a feature if its dimensionality is too high. [6] selected the most active neurons as features to avoid this problem, yet the approach discards information from a number of other neurons. To take into account the contributions of all neurons, we suggest dimensionality reduction approach. We demonstrate the predictive context information only moderately deteriorates after a degree of dimensionality reduction, by applying principal component analysis (PCA), one of the most simple and widely used dimensionality reduction technique. Figure 2 (right) shows PCA retains predictive context information significantly up to 50% reduction of dimension.

4 Conclusion

By analyzing the representation of a RNNLM, we suggest that it can be interpreted as a predictive context. Combined with the view that a hidden vector is a history summary, we conclude the vector represents a local context of a given sequence including both the past and the near future. Dimensionality reduction facilitates the practical application of the hidden vector, by fairly preserving the predictive information.

The RNNs in our experiments are relatively weak ones, because of the smaller number of hidden units and lack of novel performance-enhancing techniques. Nonetheless, we anticipate the proposed predictive context to become even stronger for more powerful RNNs since they have stronger ability to capture the sequential regularity, and hence the predictive property. Augmenting context presenting power can be an interesting future research problem. Among amendments proposed for RNNs, introducing slow-varying neurons is a promising approach, since those neurons may reflect more abstract concepts [12].

Acknowledgments

This work was supported by the ICT R&D program of MSIP/IITP (14-824-09-014) and supported in part by KEIT grant funded by the Korea government (MKE) (KEIT-10044009), NRF grant funded by the Korea government (MSIP) (NRF-2010-0017734-Videome), and the ICT R&D program of MSIP/IITP (10035348).

References

[1] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of traning Recurrent Neural Networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML13)*, pages 1310-1318, 2013.

[2] Yoshua Bengio, Nicolas Boulanger-Lewandowski and Razvan Pascanu. Advances in optimizing recurrent networks. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8624-8628, 2013.

[3] James Martens, and Ilya Sutskever. Learning Recurrent Neural Networks with Hessian-Free Optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML11)*, pages 1033-1040, 2011.

[4] Tomas Mikolov. Statistical language models based on neural networks. *Ph.D. thesis*, Brno University of Technology, 2012.

[5] Ilya Sutskever, James Martens, Geoffrey Hinton. Generating Text with Recurrent Neural Networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML11)*, pages 2017-1024, 2011.

[6] Grzegorz Chrupaa. Text segmentation with character-level text embeddings. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[7] Ozan Irsoy, and Claire Cardie. Bidirectional Recursive Neural Networks for Token-Level Labeling with Structure. *ArXiv preprint arXiv:1312.0493*, 2013.

[8] Yoshua Bengio, Gregoire Mesnil, Yann Dauphin, and Salah Rifai. Better Mixing via Deep Representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML13)*, pages 552-560, 2013.

[9] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML11)*, pages 833-840, 2011.

[10] Kyunghyun Cho. Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014.

[11] Herbert Jaeger. The "echo state" approach to analyzing and training recurrent neural networks. *German National Research Center for Information Technology GMD Technical Report*, 2001.

[12] Wataru Hinoshita, Hiroaki Arie, Jun Tani, Hiroshi G. Okuno, and Tetsuya Ogata. Emergence of Hierarchical Structure mirroring Linguistic Composition in a Recurrent Neural Network. *Neural Networks*, **24**(4):311-320, 2011.