Word2Bits - Quantized Word Vectors

Maximilian Lam
maxlam@stanford.edu

Abstract

Word vectors require significant amounts of memory and storage, posing issues to resource limited devices like mobile phones and GPUs. We show that high quality quantized word vectors using 1-2 bits per parameter can be learned by training word2vec with a quantization function. We furthermore show that training with the quantization function acts as a regularizer. We evaluate our word vectors on standard word similarity and analogy tasks and on question answering (SQuAD). Our final word vectors not only take 10-16x less space than full precision (32 bit) word vectors but also outperform them on word similarity tasks and question answering.

1 Introduction

Word vectors are extensively used in deep learning models for natural language processing. Each word vector is typically represented as a 300-500 dimensional vector, with each parameter being 32 bits. As there may be millions of words, word vectors may take up to 3-6 GB of memory/storage – a massive amount relative to other portions of a deep learning model[1]. These requirements pose issues to memory/storage limited devices like mobile phones and GPUs.

Furthermore, word vectors are often trained from scratch on application specific data for better performance in application specific domains[1]. This motivates directly learning high quality compact word representations rather than adding an extra layer of compression on top of pretrained word vectors which may be computational expensive and degrade performance.

Recent trends indicate that deep learning models can reach a high accuracy even while training in the presence of significant noise and perturbation[1]. It has furthermore been shown that high quality quantized deep learning models for image classification can be learned at the expense of more training epochs[1]. Inspired by these trends we ask: can we learn high quality word vectors such that each parameter is only one of two values, or one of four values (quantizing to 1 and 2 bits respectively)?

To that end we propose learning quantized word vectors by introducing a quantization function into the word2vec loss formulation – we call our simple change word2bits. While introducing a quantization function into a loss function is not new, to the best of our knowledge it is the first time it has been applied successfully to learning word vectors.

In this report we show that

- It is possible to learn high quality quantized word vectors which take 10x-16x less storage/memory than full precision word vectors. Experiments on both intrinsic and extrinsic tasks show that our learned word vectors perform comparably or even better on many tasks.
- Standard word2vec may be prone to overfitting; the quantization function acts as a regularizer against it.

2 Related Work

3 Word2Bits - Quantized Word Embeddings

4 Experiments and Results

5 Discussion and Future Work

Acknowledgments

References

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

- COMPRESSING WORD EMBEDDINGS VIA DEEP COMPOSITIONAL CODE LEARNING - FASTTEXT.ZIP: COMPRESSING TEXT CLASSIFICATION MODELS - FASTTEXT (Bag of Tricks for Efficient Text Classification) - Intrinsic Evaluation of Word Vectors Fails to Predict Extrinsic Performance - Improving Distributional Similarity with Lessons Learned from Word Embeddings - Optimal brain damage - Second order derivatives for network pruning: Optimal brain surgeon - Learning both weights and connections for efficient neural networks - Distributed Representations of Words and Phrases and their Compositionality - Efficient Estimation of Word Representations in Vector Space - Binarized Neural Networks - Hinton, Geoffrey. Neural networks for machine learning. Coursera, video lectures, 2012. - GloVe: Global Vectors for Word Representation - Learned in Translation: Contextualized Word Vectors - HALP - Towards Lower Bounds on Number of Dimensions for Word Embeddings - Evaluation methods for unsupervised word embeddings - SQuAD - DrQA - WordSim353 Finkelstein - WordSim Similarity, WordSim Relatedness (Zesch et al., 2008; Agirre et al., 2009 - Bruni et al.s (2012) MEN - Radinsky et al.s (2011) Mechanical Turk - Luong et al.s (2013) Rare Words dataset - Hill et al.s (2014) SimLex-999 - Low precision arithmetic for deep learning. - Predicting parameters in deep learning - Compressing neural networks with the hashing trick