Semi-Supervised Recursive Autoencoders

Adrian Guthals
aguthals@cs.ucsd.edu

David Larson
dplarson@ucsd.edu

Abstract

We evaluate semi-supervised recursive autoencoders (RAE) as a method for predicting the sentiment of sentences. Using random word initialization, we are able to predict the sentiment of a movie review dataset with a 00.0% accuracy, which is comparable to the 00.0% accuracy reported in the 2011 paper "Semi-Supervised Recursive Autoencoders" by Soch et al.

1 Introduction

Socher et al. presented a semi-supervised method for learning meanings of sentences using recursive autoencoders [1].

The lecture notes state blah [2].

Mention: neural networks, sentence meaning/sentiment

2 Recursive Autoencoders

RAE, neural networks, backpropogation, error functions, greedy algorithm, calculating derivatives numerically using finite center-difference

2.1 Error Function

$$E_1(k) = \tag{1}$$

$$E_2(k) = \tag{2}$$

2.2 Binary Tree Construction

2.3 Backpropogation

Backpropogation is an efficient method for computing the derivatives required for training a neural network. Given

2.4 Goal of Training

2.5 Gradient Verification

It is important to verify the accuracy of the gradients calculated using backpropogation. For this study we have chosen to verify the accuracy of backpropogation by comparing against gradients calculated numerically using finite central-differences:

$$\frac{\partial J}{\partial \theta} = \frac{J(\theta + \epsilon) - J(\theta - \epsilon)}{2\epsilon} + O(\epsilon^2) \tag{3}$$

where ϵ is the grid spacing.

Table 1: Number of total snippets (N_{total}), positive snippets (N_{pos}), and negative snippets (N_{neg}) for the original, training, and testing datasets.

Dataset	N_{total}	N_{pos}	N_{neg}
Original	10662	5331	5331
Training	7462	0000	0000
Testing	3200	0000	0000

3 Experiments

3.1 Datasets

We use the same movie reviews dataset as in [1], which consists of 10662 snippets from reviews posted to the Rotten Tomatoes website[1]. Each snippet is roughly equivalent to a single sentence and includes a positive/negative label, with the entire dataset containing 5331 positive and 5331 negative labelled snippets. For all experiments we randomly selected $\sim 70\%$ of the original dataset as a training set, with the remaining $\sim 30\%$ used as a testing set (see Table 1). In splitting the dataset we have taken care to prevent any snippets from existing in both sets, so as to not contaminate the results.

3.2 Optimization

We use limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), a well-known quasi-Newton optimization method, to learn the parameters θ. As we are writing our code in Python, we elected to use the L-BFGS function from the SciPy library [3].

Convergence: error less than 10^{-6} (as stated in the project description)

Regularization: ?

3.3 Experiment 1: RAE

The full method (RAE)

3.4 Experiment 2: RAE without Derivatives

RAE without derivatives to adjust the meaning vector of each word

4 Conclusion

Final remarks

References

[1] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[2] C. Elkan, "Learning meanings for sentences," http://cseweb.ucsd.edu/~elkan/250B/, February 2013.

[3] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," http://www.scipy.org/, 2013.

[1] `http://www.rottentomatoes.com`

Table 2: Words predicted to be the most positive and negative.

Ranking	**Positive**	**Negative**
1	beautiful	fails
2	brilliant	boring
3	thoughtful	neither
4	triump	bad
5	flaws	flat
6	beautifully	predictable
7	success	bore
8	spectacular	poorly
9	enjoyable	suffers
10	wonderful	unnecessary

Table 3: Phrases (length 2) predicted to be the most positive and negative.

Ranking	**Positive**	**Negative**
1	moving and	lack of
2	an enjoyable	boring .
3	and beautifully	how bad
4	a moving	the dullest
5	a triumph	flat ,
6	a beautiful	how bad
7	the best	it fails
8	and powerful	it isn’t
9	its flaws	and predictable
10	a wonderful	a boring