Deep Learning for Sentiment Analysis

A Survey of Methods

Matthew Hauser & Yating Jing

Johns Hopkins Department of Computer Science

mhauser5@jhu.edu, yating@jhu.edu

JOHNS HOPKINS
WHITING SCHOOL *of* ENGINEERING

Abstract

The application of sentiment analysis to text is a popular area of research in natural language processing and computational linguisitcs. The basic task of sentiment analysis is to identify the polarity of a piece of text. It seeks to identify the authors opinion, emotion or intent in the text. In this project we applied a deep learning model - Recursive Neural Network to text understanding and semantic analysis on the IMDB movie reviews. Each review will be classified as positive or negative. In addition, we built a logistic regression bianry classifer and trained it on the word vectors encoded from the reviews data as our baseline model, so as to better evaluate the performance of the RNN classifier.

Introduction

The task at hand is a simple binary classification, labeling a review as either positive or negative.

Our data consists of reviews from the Internet Movie Database, IMDB. These reviews are related to a specific movie and are labeled with a rating out of seven stars. We have 20,000 labeled reviews for training and 5,000 for evaluating accuracy. In addition we have 50,000 unlabeled reviews to assist with training.

In order to treat this problem as a binary classification we convert the 10 point scale to a binary label by assuming all reviews with a rating of 7 or higher are positive and less than 7 as negative.

A number of different classifiers could be used for this task, including linear regregression, support vector machines, naive bayes, etc. In order to extract context and semantic information, we chose to build a recursive neural network to better capture such meaningful attributes.

Main Objective

The main objective of this work is to classify a move review has either positive or negative. Movie reviews on IMDB are rated on a scale of 1 to 10. For our purposes a rating of 7 or higher is considered positive.

The following is an example review from IMBD for the movie, *The Dark Knight*. It received a rating of 8 out of 10 which according to our mapping is positive.

```
I have just come out of the ''Dark Curtain'' Screening for The
Dark Knight in Adelaide, and I'm blown away.  Nolan's directing
is sublime, the pacing of the movie was so well kept, managing
to keep the audience enthralled in an almost 3 hour movie.  There
were several shots in the film that had me saying ''Now how in
the hell did they do that?!'', Brilliant scores by James Newton
Howard and Hans Zimmer.  Every actor shines, no one person steals
the film.  I don't want to say much about it, because whatever
is said cannot do it justice, there is enough hype surrounding
it's release, but it's there for a reason.  Top Notch film.  Bravo
to Nolan and his crew for giving a movie that Batman truly deserves.
```

Our goal is to use this review (and many others) in training to construct a vocabulary so that we can encode word into one-hot word vectors. During classification we take a review such as this one (that is unlabelled) and attempt to determine its label.

Related Works

In [3], several machine learning applications including movie review sentiment analysis are carried out using word vectors. In the IMDB review classification experiment, a linear SVM is employed in different settings. The model utilizing bag of words representation and trained on extra unlabeled data showed superior performance.

In [2], one-hot word vectors are fed into a RNN encoder-decoder for a machine translation decoder task. The word embedding experiment conducted in the paper shows the ability of word vectors and recursive neural networks to capture the semantic similarities among words, where the words of similar semantic meanings tend to cluster together.

In [4] a detailed survey on training algorithms for neural network including back-propogation and back-propoogation through time is conducted.

The article in [1] introduces the python library Theano. Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently.

Approach

Logistic Regression

As a baseline, we trained our model using logistic regression.

Given a review as described above, we are seeking to maximize the MAP log-likelihood of the movie review and the most probable sentiment label as described in the left and right terms in the equations below.

$$\sum_{k=1}^{|D|}\sum_{i=1}^{N_k}\left(\log p(\hat\theta)+\log p(w_i \mid \hat\theta_k; R, b)\right)+\sum_{k=1}^{|D|}\sum_{i=1}^{N_k}\log p(s_k \mid w_i; R, \psi, b_c) \quad (1)$$

We use a softmax function for the word distribution $p(w \mid \theta)$,

$$p(w \mid \theta; R, b)=\frac{\exp(\theta^T\phi_w+b_w)}{\sum_{w'\in V}\exp(\theta^T\phi_{w'}+b_{w'})} \quad (2)$$

Recurrent Neural Network

The RNN classifier is trained to maximize the following conditional log-likelihood:

$$\max_\theta \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(\mathbf{s}_n \mid \mathbf{R}_n) \quad (3)$$

where $\mathbf{R}=(R_1,\ldots,R_N)$ is the movie reviews and $\mathbf{s}=(s_1,\ldots,s_N)$ denotes the output labels.

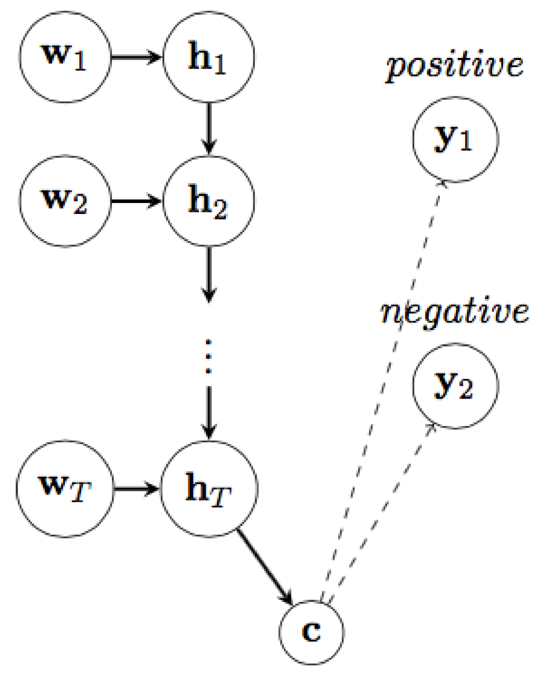

Figure 1: Model Structure

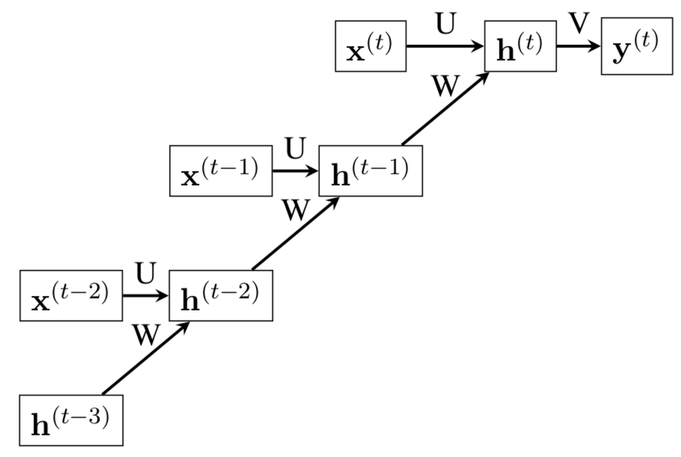

Figure 2: Illustration of one epoch of RNN as a deep feed-forward network

For recurrent neural network training, the BP algorithm is not optimal, the back-propagation through time(BPTT) training algorithm can ensure that the network will learn what information to be stored in the hidden layer.

Gradient of error vector in the output layer $\mathbf{e}_o^{(t)}$ at each time step is:

$$\mathbf{e}_o^{(t)}=\mathbf{d}^{(t)}\oplus\mathbf{y}^{(t)} \quad (4)$$

where $\mathbf{d}^{(t)}$ is the correct label vector and $\mathbf{y}^{(t)}$ is the output label vector.

In the RNN errors are propagated from the hidden layer $\mathbf{h}^{(t)}$ to the hidden layer from the previous time step recurrently as follows:

$$\mathbf{e}_h^{(t-\tau-1)}=d_h(\mathbf{e}_h^{(t-\tau)^T}\mathbf{W}, t-\tau-1) \quad (5)$$

where the error vector is obtained using element-wise function $d_h()$:

$$d_{hj}(x,t)=xh_j^{(t)}(1-h_j^{(t)}) \quad (6)$$

This unfolding process can be applied for as many time steps as many training samples were already seen, however a certain number of steps of unfolding would be sufficient since error gradients quickly vanish as back-propagated in time.

The weight matrix $\mathbf{U}$ between the input layer and hidden layer is then updated as:

$$u_{ij}^{(t+1)}=u_{ij}^{(t)}+\sum_{\tau=0}^{T}\eta\, x_i^{(t-\tau)}e_{hj}^{(t-\tau)} \quad (7)$$

where η is the learning rate. Note that $\mathbf{U}$ must be updated in one large change instead of during an incremental changing process, which can lead to instability.

The update function for the recurrent weight matrix $\mathbf{W}$ is then:

$$w_{lj}^{(t+1)}=w_{lj}^{(t)}+\sum_{\tau=0}^{T}\eta\, h_l^{(t-\tau-1)}e_{hj}^{(t-\tau)} \quad (8)$$

And the elemnent-wise update equation for the weight matrix $\mathbf{V}$ between the hidden layer and the output layer is analogous.

Results and Analysis

When training and evaluating using a logistic regression model, we obtained a baseline result of 0.79. Due to the substantial time needed for training process, we are only able to evaluate our model on a small dataset (200 reviews, 160 for training and 40 for testing). The classification accuracy for RNN model is 0.62 while the accuracy for logistic regression model is 0.59.

Forthcoming Research

The techniques used for analyzing IMDB reviews are directly applicable to many other forms of sentiment analysis. The methods could clearly be adapted for other areas in Natural Language Processing, such as machine translation, since we could use the output space to regenerate words.

References

[1] Frederic Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua. Bengio. Theano: new features and speed improvements. *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.*, 2012.

[2] van Merrienboer B. Gulcehre C. Bougares F. Schwenk H. Bengio Y. Cho, K. Learning phrase representations using rnn encoder-decoder for statistical machine translation. 2014.

[3] Andrew L. Maas. Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 1, 2011.

[4] T. Mikolov. Statistical language models based on neural networks. *Presentation at Google, Mountain View*, April 2012.

Acknowledgements

We would like to thank the Kaggle competition for inspiration for this project.