Deep Learning A3
Natural Language Processing

M&M:

Mehmet Ugurbil
Courant Institute
New York University
mu388@nyu.edu

Mark Liu
Courant Institute
New York University
ml4133@nyu.edu

Abstract

We try to use a convolutional neural network to predict the numerical star rating of a Yelp review when given only the corresponding text review.

1 Data set

We use the Yelp academic data set from the Yelp Dataset Challenge, which is a set of business reviews each labeled with a numerical star rating 1 through 5.

2 Pre-processing

We use the GloVe database to first turn every word in the review into a n-dimensional vector. Then we stack all these vectors in order to 2D form of a M by n dimensional matrix, where M is the number of words in the review. This matrix is then padded with vectors of 0s until it reaches a large fixed length K which is constant across reviews. In this way, we turn every review into a K by n dimensional matrix. We took K to be 100. If there is more than 100 words in the review, we truncate. We used Twitter-25d for our GloVe-table so that $n = 25$.

3 Simple Neural Network

3.1 Architecture

We tried several networks. First, we tried only a linear layer. This took all the 2500 inputs to a single classification.

Second, we constructed a neural network with two layers. The first layer is a convolutional layer. The second layer is linear. There is a hard tanh function in between the layers. We used a step size of 5 for the convolution. The linear layer returns one feature only, the classification.

3.2 Learning Procedure

We use the 9600 lines for training and 400 lines for validation.

Stochastic Gradient Descent was used for the learning procedure of the neural net. We used a batch size of 128, momentum of 0.1, initial learning rate of 0.001 and learning rate decay of 10^{-5}.

4 Convolutional Neural Network

4.1 Architecture

Our neural network is 5 layers. The first three layers are convolutions followed by max pooling. Then there are two linear layers. The first convolutional layer has step size 5, the second has 4, third has 2. After each convolution there is max pooling with step sizes of 5, 4, 3 in order. Each convolutional layer doubles the feature size it recieves. So we have 25 features to start with since we use Twitter 25-vector, and we get $200 = 8 * 25$ features at the end. Then we have the fully connected layer which takes in all the features and returns 16 new features. The last fully connected layer takes these 16 into 1 feature which is the classification. Both linear layers have dropout with probability 0.2. Then it is fed into a logsoftmax. We used absolute value criterion.

4.2 Learning Procedure

We use the 9600 lines for training and 400 lines for validation.

Stochastic Gradient Descent was used for the learning procedure of the neural net. We used a batch size of 128, momentum of 0.3, initial learning rate of 0.001 and learning rate decay of 10^{-5}.

5 Fast Network

5.1 Architecture

We started using 50-dimensional GloVe vectors. So, $n = 50$ in preprocessing. Also, if there are less than $K = 100$ words in the input, then we use the same words to fill up the matrix.

The convolutional network consists of two layers, one convolutional and one linear layer. There is a hard tanh function in between the layers. The linear layer feeds to a log soft-max function. Negative log likelihood criterion is used to pick from the 5 outputs.

We also built a SQL data-base for faster lookup of the GloVe table.

5.2 Learning Procedure

We use the 9600 lines for training and 400 lines for validation.

Stochastic Gradient Descent was used for the learning procedure of the neural net. We used a batch size of 128, momentum of 0.01, initial learning rate of 0.03 and learning rate decay of 10^{-2}.

6 Results

We get error of 0.72 with the simple layer. We get 0.8 error with the large convoltional network which suggests that it is not working. For the fast network, we get 0.62 error and super fast runtime so it is the best out of all of them.

References

[1] LeCun, Y., Bottou, L., Genevieve B.O., & Muller K.R. (1998) *Efficient BackProp* New York: Springer.

[2] Zhang, X., LeCun, Y. (2015) *Text Understanding from Scratch*, ArXiv.

[3] Collobert, R. et al (2011) *Natural Language Processing (almost) from Scratch*, ArXiv.