News Article Revenue Prediction Using Deep Learning

Joshua Griffin, Levi Lian, Kayiita Johnson
{jrtg, levilian, kayiita@stanford.edu}

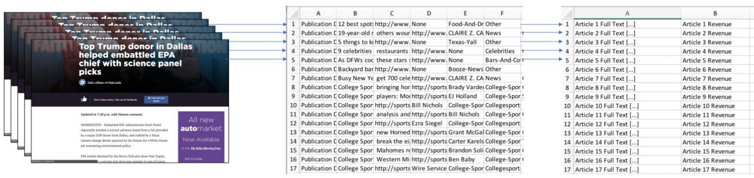
deepnews.ai

Overview

News media companies are facing increasing pressure to decrease costs and boost revenues. At the same time, news readers are desiring more news articles that they care about from their media outlets. We use deep learning to help address these mutual needs by using full news article text to predict advertising revenue generated from a given article webpage. To make our predictions, we use CBOW-generated, GloVe and Fasttext word embeddings, two-layer LSTM and feed forward DNN to build separate models. Our testing showed that the LSTM architectures produced the highest accuracy results which we attribute to its enhanced ability to distinguish language context and use past and future information.

Data & Features

Our data consists of 45,000 online news articles and their features from the Dallas Morning News website and the revenue generated from each article webpage after a 2-week period. Our input data is in CVS format with article text and associated features including revenue data on a single line. We split our data into 80%/10%/10%, training, development, and testing data sets respectively.

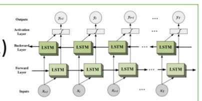


While our data features include information on publish dates, reader engagement levels, referring websites, user device types, and more, we focus on the text content of each article and the associated revenue that was generated. This allows us to probe the link between editorial content and revenue generation to further our long-term goals of increasing article content quality, article topic desirability, and news media revenue generation.

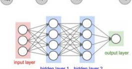
Model Selection

We implement two base models: (1) is a feed-forward, deep neural network based on research by Ding et al. where simple networks are able to achieve high performance on a similar problem; (2) is a uni-directional LSTM network based on Stokowiec et al. where the networks were able to overcome the vanishing and exploding gradient problem experienced for long text inputs while contextualizing in both directions.

(1)



(2)



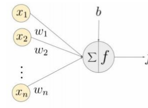
References

- Xiao Ding, Yue Zhang, Ting Liu, Junwen Duan: Deep Learning for Event-Driven Stock Prediction, 2015
- Wojciech Stokowiec, Tomasz Trzcinski, Krzysztof Wolk, Krzysztof Marasek and Przemyslaw Rokita: Shallow reading with Deep Learning: Predicting popularity of online content using only its title, 2017
- Tang D, Qin B, Feng X, and Liu T: Effective LSTMs for target-dependent sentiment classification, 2016.
- Leonardo dos Santos Pinheiro, Mark Dras: Stock Market Prediction with Deep Learning, 2017

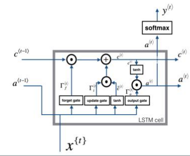
Quantitative Description

The output of each DNN neuron is based on a linear combination of all previous layer outputs and the respective scaling weight and constant offset for each node. Finally, the inputs are summed and transformed by a non-linear activation function (f) before being passed to the next layer as shown below. Similarly, the uni-directional, LSTM is composed of individual cells. Each cell accepts input from the current and all previous time-steps via the previous LSTM cell (zero-padding applied). Each cell also produces an output based on the inputs and set of memory control gates with parameters trained as shown below. We use the logarithmic and cross entropy loss function shown below to minimize the cost.

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log p_{ij}$$



$$f\left(b+\sum_{i=1}^{n} x_i w_i\right)$$



$$\Gamma_f^{\langle t\rangle}=\sigma(W_f[a^{\langle t-1\rangle},x^{\langle t\rangle}]+b_f)$$
$$\Gamma_u^{\langle t\rangle}=\sigma(W_u[a^{\langle t-1\rangle},x^{\langle t\rangle}]+b_u)$$
$$\tilde{c}^{\langle t\rangle}=\tanh(W_C[a^{\langle t-1\rangle},x^{\langle t\rangle}]+b_C)$$
$$c^{\langle t\rangle}=\Gamma_f^{\langle t\rangle}\circ c^{\langle t-1\rangle}+\Gamma_u^{\langle t\rangle}\circ\tilde{c}^{\langle t\rangle}$$
$$\Gamma_o^{\langle t\rangle}=\sigma(W_o[a^{\langle t-1\rangle},x^{\langle t\rangle}]+b_o)$$
$$a^{\langle t\rangle}=\Gamma_o^{\langle t\rangle}\circ\tanh(c^{\langle t\rangle})$$

Hyperparameter Tuning

To increase algorithm learning efficiency, we use Adam optimization as described by Kingma et al., which combines stochastic gradient descent and RMS propagation. We would loop over multiple combinations of learning rates, epochs and batch sizes. Currently, we have the values of these hyperparameters set to: alpha = 1e-3, beta1 = 0.8, beta2 = 0.999, and epsilon = 1e-8. We would tune the number of epochs to its final value by iterating. We would use 128 time-step along with a feedforward layer at the output to evaluate the model performance.

We created the models for the feed-forward, deep neural network with word embeddings, as well as the uni-directional LSTM model. We were able to go through one epoch for the feed-forward model, which generated training error of 66%. We were unable to go through epochs for the LSTM model, as each epoch would take us seven hours using AWS. Though we were not able to test this model on the test data, it does seem possible to predict article revenue generation from article content and that doing so could significantly increase the ROI of new news articles for news media companies.

Results & Discussion & Future

This implies that there is a strong link between article content, reader click-through rates, and by extension, reader views and satisfaction. To the best of our knowledge, this is the first application of DL to this problem. For future work, we recommend using additional ways to make the algorithms more efficient, given the duration of our training lengths.

Model	Training Error	Test Error	train set size	test set size
FF-DNN w/ CBOW			25,530	3,192
FF-DNN w/ Word Embeddings	66%		25,530	3,192
Uni-directional LSTM			25,530	3,192
Human				

We would also recommend using a modular approach that separates sentiment context, location of the news in the news agency main page, and behavioral analysis to shed light inside the DL "black box" to see better what types of factors are driving correlation we saw. Specifically, we would explore the efficacy of Attention and stacked architectures if information can ultimately help bring value and trust back to the news media industry.