Make AI Great Again

CS230

Naveen Srivatsa
nns@stanford.edu
Stephanie Scott
smscott@stanford.edu

Generating Tweets in 'Presidential' Style Using Recurrent Neural Networks

Abstract

President Donald Trump's Twitter account has been noted for its distinctive writing style. We attempt to build a recurrent neural network that trains on 2,829 @realDonaldTrump tweets in order to generate new tweets in the style of the president. A neural network that trains on words rather than characters and consists of 4 bidirectional LSTM layers generates tweets that closely resemble @realDonaldTrump's tweets. The addition more layers or dropout regularization has some minimal impact on the quality of the output.

Data & Features

The dataset consists of 2,829 tweets issued by the @realDonaldTrump Twitter account between March 4, 2017 and May 31, 2018. Retweets are excluded. Replies by @realDonaldTrump are included. Tweets were extracted using the TwDocs tool. Some pre-processing was required: President Trump uses a significant number of ampersands that were extracted as "\&amp" and therefore were manually replaced with the proper character. Pre-processing and normalization required for training varied by model.

Models

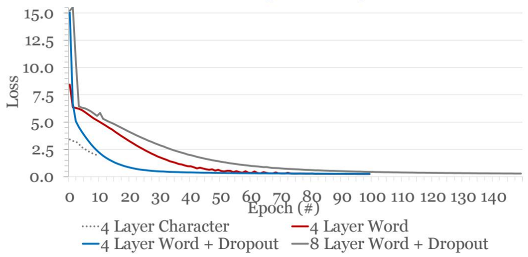
The **Character Based Model** uses a tokenizer that converts each character into an integer and is then normalized. Next, the base model applies one LSTM layer containing 256 LSTM units, applies dropout regularization, and applies a softmax activation layer. Adam optimizer is used, with learning rate of 0.001, beta1 of 0.9, and beta2 of 0.999.

The **Word Based Model** uses a tokenizer to turn each word into an integer corresponding to a token in a dictionary. Next, the model feeds embeddings into two sequential LSTM layers that contains 128 LSTM units each. The embedding and outputs of both LSTM layers are concatenated and fed into an attention layer that calculates the weighted average of inputs over time and outputs a context vector. Last, a dense layer applies a softmax activation. RMSprop is used with with learning rate of 0.004.

Results – Model Performance

Model	Description	Loss	Sample output
Character: Base model	1-layer LSTM with dropout, learning rate = 0.001	1.8479 (after 96 epochs)	uic oi the resuic oi the resuic oi the resuic oi the resuic oi the resuic oi the resuic oi the resui
Character: Base + Faster learning	1-layer LSTM with dropout, learning rate = 0.01	2.0593 (after 19 epochs)	people of the fake news the searet wery searri the people of the fake news the searet wery searri t
Character: Two-layer	2-layer LSTM with dropout, learning rate = 0.001	1.5376 (after 50 epochs)	and the fake news media is a great honor to welcome the united states and the people of the world fo
Character: Base + bidirectional	1-layer bidirectional LSTM with dropout, learning rate = 0.001	1.9569 (after 36 epochs)	the sireate niane on the uarl and seaurity oo the sases and secori big teey io the uasling onft to
Character: Two-layer + bidirectional	2-layer bidirectional LSTM, learning rate = 0.001	1.3889 (after 26 epochs)	the fake news media is a great state of the fake news media is a disaster
Word: Base model (on characters)	4-layer, 128-cell Bidirectional LSTMs, learning rate = 0.004	1.9735 (after 10 epochs)	A thateres', my of ranto the w/ Yomil resut dime the we the secuse, GRIO
Word: Base model	4-layer, 128-cell Bidirectional LSTMs, learning rate = 0.004	0.2435 (after 100 epochs)	general john kelly is doing a great job as chief of staff . all americans strong on mexico on this our country vote for against the next administration . if no support !
Word: Base model + dropout	4-layer, 128-cell Bidirectional LSTMs, learning rate = 0.004, dropout = 0.2	0.2577 (after 100 epochs)	steve bannon will be interviewed by @ seanhannity at 9pme on @ foxnews
Word: Base model + dropout + 8 layer	8-layer, 128-cell Bidirectional LSTMs, learning rate = 0.004, dropout = 0.2	0.2875 (after 150 epochs)	michael wolff is a total loser who made up stories in order to sell this would go up classified information ! spending !

Results - Loss under varying architectures and parameters for model with multiple LSTM layers



Discussion

Character vs Word Model – A multi-layer character model can reproduce words and phrases that resemble President Trump's tweets (e.g., "fake news"). Character-based model is prone to spelling errors but captures capitalization. Word-based models are superior, as they can produce coherent sentences or tweets.

Learning Rate – Increasing the learning rate resulted in no meaningful decrease in the runtime of each epoch.

Dropout – Adding dropout to the 4-layer model improved performance and loss convergence slightly.

Bidirectional layers – Replacement of the feedforward layers with bidirectional layers in the two-layer character-based models reduces the number of epochs required to achieve low levels of loss but increases the length of each epoch.

Exploding gradients – Character-based models with faster learning or bidirectional layers see evidence exploding gradient. Future versions of model should incorporate gradient norm clipping.

Overfitting – Some output tweets closely resemble input data (e.g., "Michael Wolff is a total loser who made up stories in order to sell"). Future versions of model should reduce levels of overfitting.

Future

We find that a neural network that trains on words rather than characters and consists of 4 bidirectional LSTM layers generates tweets that closely resemble @realDonaldTrump's tweets. Future iterations of this model should begin with the word-based model and focus on reducing levels of overfitting.

References:
Jaiswal, S. (2017). Keras Recurrent Neural Network with Python. GitHub repository, https://github.com/sagar448/Keras-Recurrent-Neural-Network-Python
Pascanu, R., Mikolov, T., & Bengio, Y. (2013, February). On the difficulty of training recurrent neural networks. In International Conference on Machine Learning (pp. 1310-1318).
Woolf, M. (2017). Textgenrnn. GitHub repository, https://github.com/minimaxir/textgenrnn