TITLE

Artificial poem generation with RNN style training.

AUTHORS

Anshul Gupta A20369831
Prateek Agrawal A20358932

PROBLEM DESCRIPTION

We are trying the evaluate the effectiveness of Recurrent neural networks to generate automated text. For this we will be using large amount to data scrapped from different websites online. We then feed this data character wise to the RNN network and training it to predict the next character.

DATA

Data will be collected in the form of limericks and poem by scrapping different popular websites such as **poetrysociety.org.nz** and **oedilf.com.**
For the purpose of this milestone we have collected 5000 limericks and stored them in a csv file. This data will be used to train a small deep network to test the Recurrent neural network created by us. After this successful attempt we plan to download 100000 limericks and poems and train a bigger network on the new data.

METHOD

We will be designing a deep learning network to train thousands of data on the character level. We then use this trained model to generate artificial data given a topic by the user.

Recurrent Neural Network with LSTM cells instead of the traditional RNN cell will be used. Each LSTM cell would have 500 neurons in it. We will start evaluating our model with sequence length of 25 character and keep increasing it till the time the artificial poems start making sense.

The network will be designed using the Tensor Flow library available to train neural network. The whole process of training would be done on an EC2 instance from Amazon Market Place. We have successfully install CUDA toolkit and CUDA on one such instance.

To evaluate our model, we will be creating a base line model similar to the trump bot created by the professor in the class. We would compare these two models and check which model generates data that makes more sense for the English language. Also, the model should not just be repeating the poems in the training but generated text which are entirely new.

INTERMEDIATE RESULTS

We have used beautiful soup to web scrap several websites and collect 5000 limericks. Also, we have successfully rented an EC2 instance on the amazon web services and install CUDA tool kit, CUDA and GPU version of tensor flow on it. We plan to create the RNN network next and test it on our local system for small data set. After this successful implementation we would train the model on EC2 GPU instance for a larger network and dataset.

RELATED WORK

- https://homes.cs.washington.edu/~yejin/Papers/emnlp16_sonnet.pdf
- http://nlp.stanford.edu/courses/cs224n/2015/reports/1.pdf
- https://www.aclweb.org/anthology/D/D15/D15-1044.pdf
- http://publications.lib.chalmers.se/records/fulltext/245146/245146.pdf
- http://karpathy.github.io/2015/05/21/rnn-effectiveness/

WHO DOES WHAT

Anshul Gupta

- Collecting the data from different web sited using Beautiful soup
- Creating the Baseline model discussed above.
- Evaluated the baseline model and RNN model.
- Tweak the RNN model to get the better result.

Prateek Agrawal

- Renting a EC2 instance from Amazon web services and installation of all the required modules for GPU such as CUDA and Tensor Flow.
- Construction of the RNN network with LSTM cell, each with 500 neurons and sequence length of 25.
- Training the created network with the large data set.
- Generate automated text for a given topic.

TIMELINE

- April 10th – Collection of minimum of 10000 data from poetry websites using Beautiful soup. Construction of the RNN network and the baseline model.
- April 20th – Training the RNN network on the data set.
- April 23th – Evaluating the baseline model and the RNN model.
- April 25th – Completion of the report with our findings.

REFERENCES

- http://karpathy.github.io/2015/05/21/rnn-effectiveness/
- http://sballas8.github.io/2015/08/11/Poet-RNN.html
- https://en.wikipedia.org/wiki/Recurrent_neural_network
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/