CS419: Introduction to Machine Learning

Project Report

LSTM Wordsworth

Siddharth Agarwal P17109

Prof. Preethi Jyothi
Computer Science and Engineering
IIT Bombay

Contents

1 Motivation

A lot of work has been done over the past few decades in the domain of text generation. The primary targets for this domain have been based in prose for obvious reasons. However, poetry is a domain of literature that deserves the reverence it receives. Poetry is a way to understand how language and symbol systems work. It is a worthy expression of emotion, or deep feelings, and aesthetics. Therefore it felt appropriate to attempt to generate poetry based on the work of possibly the most well-known poet of the old Romantic generation, William Wordsworth.

2 Background

Ever since Mikolov et al. demonstrated how to generate text using RNNs, neural language modelling has received a fever pitch of attention. Neural text generation has been thoroughly explored, with teams even coming up with algorithms that can amp up the emotional aspect of poetry (Misztal et al.). One of the most interesting contributions here came in the form of a blog post by Andrej Karpathy, who demonstrated using RNNs to generate Shakespearean poetry. Multiple generative approaches have also seen success in the domain.

3 Objective

The initial objective here was to compare multiple discriminative and generative approaches to generate poetry and compare them via quantitative metrics such as perplexity and accuracy, and qualitatively, by asking people with academic experience in Literature about whether they would believe that the snippets presented to them were really from Wordsworth.

However, I couldn't finish the generative approaches in time for the report. So for this project, I have attempted an analysis that compares the currently trendy recurrent neural networks architectures.

4 Dataset

The dataset here is the first volume of Shakespearean poetry out of an eight volume collection available on Project Gutenberg.

4.1 Attributes of the Dataset

Attribute	**Values**
No. of lines	10902
No. of word tokens	91093
Size of Vocabulary	9940
No. of Individual Characters	46

5 Algorithms

5.1 Preprocessing

- The data was split into training, validation and test sets in the ratio 60:20:20.

- Attributes 'race', 'therapy' and 'type of cancer' were one hot encoded.
- Attributes 'TNM staging, T' and 'TNM staging, N' are graded attributes, so they are converted to numbers 1,2,3,4 and 0,1,2,3 respectively.
- All the columns are normalized as $X_{norm} = \frac{X - \text{mean}(X)}{\text{std}(X)}$

5.2 Character Level LSTM

Number of dimension in cluster	16
Number of cluster	20
Mean square error	0.5211
R^2 value	0.6493

5.3 Character Level GRU

As the cluster is difficult to analysis in 16 dimension so the plot of the cluster in two dimension by plotting two of its major components is -

The color shows all the points in a same cluster.

5.4 Word Level GRU

Memory error due to large size of data size.

5.5 Word Level LSTM

Epsilon	0
Penalty parameter C	1
R^2 value	0.28

5.6 Neural Networks

Number of hidden layers	2
Number of units in layers	48, 16
Dropout regularization in both layers	0.1
Activation	ReLu
Optimizer	Adam
Batch Size	100
Loss	Mean squared error
Epochs	15
R^2 value	0.567

5.7 Linear Regression

Implemented basic linear regression with the Stochastic Gradient Descent algorithm and tuned the learning rate to obtain the highest R^2 value possible with no regularization.

R^2 value obtained = 0.438

5.8 Random Forest Regression

Number of trees in ensemble	30
Max depth of trees	8
Cross validation	5 fold
R^2 value	0.489

5.9 HMM Regression

We tried to implement an HMM Regression algorithm to add a generative approach to the regression problem as well but faced multiple hold ups while calculating the forward and backward probabilities and hence did not take it to completion.

6 Conclusion and Future Work

During our journey of collecting data and understanding and interpreting it, we got in touch with quite a lot of people who were actively involved in similar fields ranging from Project staff and medical students to professors working on building solutions for problems in related fields (like telepathology) through whom we learned a lot about the current scenario of such projects.
These interactions led us to the following conclusions :-

- We could use a similar approach like we did with SEER to obtain data in the Indian context from the Tata Memorial Hospital which would ideally be non-curated and could develop into a valuable project
- To tackle the problem of the genome sequence data being too large and hence having low workability, we could devise an algorithm which enables us to read and work on sections of the whole data (in the form of sentences instead of whole documents) and then implement our desired approach to identify the probability of cancerous mutations in that section
- We could expand our context to include images of tissue slices and work to flag abnormalities in these images to make the identification process faster for pathologists who look at enormous amounts of these images

7 References

7.1 SEER

- Request for accessing data
- SEER*Stat Installation
- Tutorial for accessing data

7.2 The 1000 Genome Project

7.3 Others

- Prediction of cancer causing missense variants
- Liquid Biopsies for Cancer Genetics
- HMM Regression for Life Expectancy Prediction