Lyrics Generation Using Echo State Networks

by

Shikhat Karkee

Bachelor Thesis in Computer Science

Prof. Herbert Jaeger

Name and title of the supervisor

Date of Submission: May 16, 2018

Jacobs University — Focus Area Mobility

With my signature, I certify that this thesis has been written by me using only the indicated resources and materials. Where I have presented data and results, the data and results are complete, genuine, and have been obtained by me unless otherwise acknowledged; where my results derive from computer programs, these computer programs have been written by me unless otherwise acknowledged. I further confirm that this thesis has not been submitted, either in part or as a whole, for any other academic degree at this or another institution.

Signature   Place, Date

Abstract

Lyrics have always been a crucial part of songs. The lyrics of almost all the songs are written by a human being. Depending on how fluent the human being is at song-writing, it can be a very tedious and time-consuming task. Lyrics Generation refers to an automatic generation of lyrics using some machine learning model in a relatively shorter time. Long Short-Term Memory (LSTM) model and Markov Processes are two of the widely used techniques for lyrics generation.

This project aims at finding out how Echo State Networks (ESNs) are suited best for lyrics generation and does not intend to beat the state of the art lyrics generator or provide an alternative method to it. An ESN is trained with lyrics of some love songs, and later the same ESN is used to self-generate lyrics which will be of 200 symbols in length. The quality of the network is also assessed and optimized, such that the values for the parameters being used by the ESN are optimal.

Contents

1 Introduction

Lyric generation is a complicated version of normal text generation. This is because one must generate sensible, poetic text [6] on a specific topic, as, for example, love and sorrow. Human lyrics generation can be done using two approaches, namely, the tune satisfying approach and the theme satisfying approach [13]. The tune satisfying approach deals with writing the lyric based on a given tune and melody, whereas the theme satisfying approach deals with writing the lyric based on a given scenario. The problem with the tune satisfying approach is that the semantics and the context of lyrics are compromised since the tune is given importance. On the other hand, the problem with the context-based approach is that the lyrics are given more importance, and some words in the lyrics need to be revised to fit the tune [14]. Another problem with human lyrics generation is the time-consuming aspect of it. Lyrics Generation using machine learning techniques provide a good alternative to human lyrics generation. The training of recurrent neural networks (RNNs) is a machine learning technique, which has been used by many researchers for text symbol prediction and generation tasks. [1] is one of the classical work on text symbol prediction tasks and [2] is another example in the same field. [3] is a work on text symbol generation using Echo State Networks.

Many researchers have used lyrics for different applications. [8] deals with genre classification based on feature analysis using rhyme patterns, pronouns, prepositions, digits and exclamation marks. The classification task was performed using Naive Bayes classifier, Support Vector Machines (SVM) and Decision Trees. For all three classifiers, the classification done with the feature sets introduced in the paper outperformed the approach in which only bag-of-words model was used. The shortcoming of this paper is the lower classification accuracy in comparison to other results of genre classification based on audio content. [9] is a work on topic detection of song lyrics using text mining techniques and clustering.

Lyrics Generation using constraints like rhyme structure and length was done in [7]. The lyrics were generated using Markov Processes with Constrained Markov technique. Syntactic correctness and semantic relatedness of the lyrics were the two quality assessment criteria. The procedures to assess the quality of the technique were the evaluation and comparison of the two criteria between the lyrics generated from this technique and lyrics generated from pure Markov approach without any constraints. Also, the comparison was made with lyrics generated from pure constraint solving approach using control constraints. Lack of reliable method to automatically evaluate the two criteria was a shortcoming of this paper because empirical evaluation by human ratings on the two criteria was done. The Constrained

Markov approach performed better in both comparisons. Rap lyrics similar in style to a target artist were generated using Long Short-Term Memory (LSTM) model in [10]. It compares the model with a simple n-gram model and outperforms it. The similarity of the generated lyrics to the lyrics in the training set and rhyme density were used as the quality assessment criterion. The algorithm proposed in [15] was used for similarity measurement and lyrics with lower similarity were said to be novel. The model can be considered to be good because the generated lyrics had high rhyme density and low similarity.

Echo State Network is an artificial recurrent neural network that provides an architecture and supervised learning principle for RNNs [4]. The basic idea of ESN is to use a large ”reservoir” RNN as a supplier of interesting dynamics from which the desired output is combined [3]. This reservoir is the recurrent layer formed by a large number of sparsely interconnected units with non-trainable weights. Under certain conditions, the traces of inputs from initial stages can be seen in the output of later stages. An example of ESN layout can be seen in Figure 1.

This project report is organized as follows.

Section 2 presents the description of the researched problem by viewing ESN and its formalism, as described in other articles. It also states the motivation behind the research and its objectives. Section 3 provides an insight into the data that was used, setup of the network, training procedure of the ESN reservoir, testing procedure and generation procedure. Section 4 shows us the results that were obtained from the project. Section 5 provides a discussion of the results, summary of the project and useful starting points for future works.

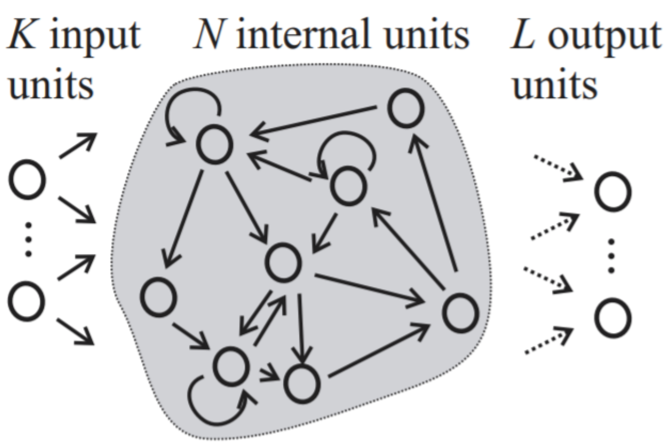

Figure 1: Echo State Network Layout. The picture was taken from [3].

2 Statement and Motivation of Research

2.1 Echo State Network

This section provides a general structure of ESN as provided in [3] and [4].

A randomly generated RNN, namely a reservoir, is the core part of the ESN. It can be seen as a nonlinear high-dimensional expansion of input signal. At the same time, it serves as a memory, providing temporal context [16].

A basic discrete-time neural network with K input units, N neurons and L output units is considered. f is a sigmoid function (a tanh function was used in this project), **W** is an $N \times N$ reservoir weight matrix, $x(n)$ is an N-dimensional reservoir state, $\mathbf{W}^{in}$ is an $N \times K$ input weight matrix, $\mathbf{u}(n)$ is a K-dimensional input signal, and **b** is an N-dimensional bias vector.

In order to harvest the states, the dynamical reservoir is driven with the training data for times $n = 1, ..., n_{max}$, where n_{max} is the total number of data. The state update equation for the activation of internal units is given by

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{b}). \tag{1}$$

The extended system state is obtained by the concatenation of the reservoir and input states at time n. It is given by: $\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]$ [4].

The output is obtained from the extended system state by [4]

$$\mathbf{y}(n) = g(\mathbf{W}^{out}\mathbf{z}(n)). \tag{2}$$

Here, g is an output activation function (typically the identity or a sigmoid) and $\mathbf{W}^{out}$ is an $L \times (K + N)$-dimensional matrix of output weights.

In state harvesting stage of training, a sequence of extended system states $\mathbf{z}(1), ..., \mathbf{z}(n_{max})$ is yielded. The obtained extended system states are filled row-wise into a state collection matrix (S) of size $n_{max} \times (N+K)$. The desired outputs $\mathbf{d}(n)$ are sorted row-wise into a teacher output collection matrix (D) of size $n_{max} \times L$ [4].

Ridge regression is used to calculate the desired output weights $\mathbf{W}^{out}$. As introduced in [5], the ridge regression is given by

$$\mathbf{W}^{out} = ((SS' + \alpha^2 I)^{-1} S' D)'. \tag{3}$$

where α is a regularizer value and I is an identity matrix of size $n_{max} \times n_{max}$.

The spectral radius (λ) of **W**, having the eigenvalues δ_1,...,δ_n, is given by [12]

$$\lambda = max\{|\delta_1|, ..., |\delta_n|\}. \tag{4}$$

W is divided by λ and then multiplied by a scaling factor. This is done in order to ensure the echo state property. The spectral radius should be greater in the tasks requiring longer memory of the input [17].

The formal learning criterion is to reduce the mean squared error. Mean squared error is given by

$$\textbf{MSE} = \frac{1}{n_{max}} \sum_{n=1}^{n_{max}} (\mathbf{d}(n) - \mathbf{y}(n))^2 . \tag{5}$$

2.2 Research Objective

I state that this project does not intend to beat the state of the art lyrics generator or provide an alternative method to it. It aims at finding out how ESNs are suited best for lyrics generation.

The main objectives of the project are:

1. To train a reservoir of ESN with a large text dataset containing lyrics of multiple love songs.
2. To assess the quality of the trained reservoir of ESN and optimize it to get optimal values for the parameters used by the ESN.
3. To use the trained reservoir with optimal parameters to self-generate sensible lyrics of specific text length (200 symbols in this project).

3 Description of the Investigation

The following section provides details on practical implementation that was done to address the research objectives listed in Section 2.2.

3.1 Data description

Lyrics of 15 love songs from various artists were used as the training dataset. Lyrics of specific songs were taken from the website `www.genius.com` and the data set provided by Sergey Kuznetsov in Kaggle [11]. The Kaggle dataset was originally acquired from `www.lyricsfreak.com`. All inconvenient data like non-English lyrics, extremely short or extremely long lyrics and lyrics with non-ASCII symbols were removed. The dataset consists of Artist, Song Name, Lyrics and the link to a web page with the song.

The genres of the 15 selected songs whose lyrics were used are pop, classic rock, soul, r&b, jazz, indie rock and electro house. The lyrics were taken from songs of the artists ranging from the old era (consisting of artists like Frank Sinatra and Elvis Presley) to the new era (consisting of artists like Ed Sheeran and Adele).

The lyrics were put into a text file and spaces were replaced by underscore "_", line breaks were replaced by forward slash "/" and paragraph breaks were replaced by backward slash "\". Even though the text had both uppercase and lowercase letters, it was ensured that all the uppercase letters were converted into lowercase letters while being used in the implementation. This resulted in 32 different symbols consisting of letters a to z, comma ",", question mark "?", single quotation mark "'", underscore "_", forward slash "/" and backward slash "\". The total text sequence was of length 15531. A summary of the data is given in Appendix A.

3.2 ESN Setup

ESN reservoir network consisting of 400 neurons was set up initially. For this, a reservoir weight matrix (**W**) of size 400×400 was sampled from a uniform distribution over [-0.5, +0.5]. Similarly, a weight matrix for input ($\mathbf{W}^{in}$) of size $N \times K$ and bias (**b**) of size $N \times K$ were generated. The spectral radius (λ) was calculated by taking the maximum absolute value of the eigenvalue of the reservoir weight matrix. The weight matrix of the reservoir was divided by the spectral radius (λ) in order to ensure the echo state property. Also, the scaling factor variables (wsf, $winsf$ and bsf) were initialized by choosing a value randomly. The values were chosen as 1 for all three variables. These were multiplied with **W**, $\mathbf{W}^{in}$ and **b** respectively.

3.3 Training

The echo state network was equipped with 32 input units and 32 output units with no feedback connections. The position for the input units and output units were assigned such that “a” was assigned the 1st input unit, “b” was assigned the 2nd input unit, and following this pattern, “z” was assigned the 26th input unit. 27th to 32nd input units were assigned in the order of “,”, “_”, “?”, “”, “\”, “/”, excluding the double quotation marks (“”). For the initial training of ESN, 15531-symbol long text sequence was used. Each symbol was coded as input signal such that the input unit assigned to that symbol was set to 1 and the rest was set to 0. For example, if the symbol was “a”, then

$$\mathbf{u}(n) = (1, 0, 0, ...0)'. \tag{6}$$

Only one symbol per network update cycle was fed. For teacher signal, similar encoding of the next symbol was used at the output unit such that the unit assigned to that specific symbol was set to 1 and the rest was set to 0. For example, if the next symbol was “b”, then

$$\mathbf{d}(n) = (0, 1, 0, ...0)'. \tag{7}$$

Thus, in this manner, the reservoir network learnt to predict the next symbol.

3.4 Testing

The following section focuses on the quality assessment procedure of the ESN and its operational check using Markov Chain.

3.4.1 Optimization

5-fold cross-validation scheme was used to split the 15531 symbols into training and validation sets. Entropy rate was taken as the criterion to assess the quality of the trained ESN. During each loop of cross-validation, output weight matrix obtained from training was used to calculate the output vector. The output vector was turned into a probability vector by turning the negative entries to 0 and normalizing, such that they sum to unity. Entropy rate was calculated using this probability vector, as the mean of the logs of the probability number given by the predictor for the actually occurring observation. After the end of validation run, mean entropy rate was calculated from the 5 entropy rates obtained during each validation run. The network scaling parameters (wsf, $winsf$ and bsf), the ridge regression regularizer (α) and the number of neurons (N) (that were initialized with a randomly chosen value) were then changed, such that the mean entropy rate was high. High entropy rate denotes that, during generation, the probability of the generated symbol is closer to the actual probability.

3.4.2 Operational Check

This operational check was done in order to check the correctness of the implementation. A text generator using Markov chain was implemented. It was used to generate 1000 symbols long synthetic training data (pseudo-text) with 2 simple alphabets (a and b). The generated synthetic training data is given in Appendix B. The generated data was then fed into the ESN to predict probabilities of the next symbol. The procedure as described in Section 3.3 was followed in order to train the ESN. The predicted probabilities were checked against the known probabilities from the transition matrix of the Markov Chain, and they were very near to the corresponding probabilities in the transition matrix. This means that the ESN was able to train properly without any bug.

3.5 Generation

After parameter optimization with cross-validation, the network was first re-trained using the entire 15531-symbols long text sequence with the updated optimal scaling parameters and regularizer value. The same procedure as in training was followed.

Then, the network was used to generate a text sequence of lyric of length 200 on its own when the first symbol was given. In order to generate initial network state, the network was fed with 20-symbols long correct lyric. The lyric was used from one of the songs that was part of the training data. The weight of the output matrix ($\mathbf{W}^{out}$) was used as computed during training in order to calculate the output vectors. ($\mathbf{y}_1(1), ..., \mathbf{y}_{32}(1)$) was the output vector for the first generated symbol. The negative entries of the calculated output vector was set to 0 and normalized, such that the sum of the entries was 1, converting it into a probability vector. ($\mathbf{p}_1(1), ..., \mathbf{p}_{32}(1)$) was the probability vector for the first generated symbol.

The probability vector was raised to F-th power and re-normalized, such that the sum of the entries was 1. In order to select the first generated symbol from this probability vector, a weighted random draw of the index of the probability vector was done. Probability values at the corresponding index of the probability vector were used as the weights for the draw. Also, this was fed back into the input unit by setting the input unit assigned to that symbol to 1 and the rest to 0. Then, the same procedure as discussed above was used to get the next output vector and the probability vector for the next symbol prediction. This procedure was iterated until a text sequence of specific desired length (200 in this project) was generated.

4 Results

50 step output traces during final training for symbols “a”, “z”, “_” and “o” can be seen in Figure 2. From the figure, it can be seen that negative output values occur. So, normalization of this vector during generation is necessary in order to convert it into a probability vector. Reservoir activation during final training can be seen in Figure 3. It can be seen that different neurons have different change in activation values, denoting that the network is not deterministic. During parameter optimization, the values of the parameters were changed which changed the values of the entropy rate as well. The changes found were as follows:

1. The mean entropy rate increased when the number of neurons was increased from 100 to 400. However, the increase was small for the number bigger than 400. So, 400 was chosen as the optimal size of the reservoir.

2. The mean entropy rate increased by higher value when the value of α was decreased in the range of $[0.005, 1]$. However, it started increasing with minimal value for $\alpha < 0.005$. So, 0.005 was chosen as the optimal ridge regression regularizer value.

3. The mean entropy rate increased when the value of wsf was used in the range of $[1, 1.4]$, but it decreased when the value of $wsf > 1.4$. So, 1.4 was chosen as the optimal scaling factor for reservoir weight matrix.

4. The entropy rate increased when the value of $winsf$ was used in the

range of $[1, 1.6]$, but it decreased when $winsf > 1.6$ was used. So, 1.6 was chosen as the optimal scaling factor for input weight matrix.

5. The entropy rate increased when the value of bsf was used in the range of $[0.1, 0.9]$, but it decreased when $bsf > 0.9$ was used. So, 0.9 was chosen as the optimal scaling factor for bias matrix.

The optimal values for the parameters after quality assessment of the trained ESN can be found in Table 1.

Parameters	**Values**
N	400
α	0.005
wsf	1.4
$winsf$	1.6
bsf	0.9

Table 1: Optimal values of parameters after quality assessment of the trained ESN.

Before starting free generation, the network was fed with 20-symbols long correct lyric. It was then let to generate lyric having a length of 200 symbols freely. The following 200-symbols long lyric was generated using F = 1:

```
look_at_her_face,_ith_iein_ahyos'nccd_udahnh_yha_unwp_hsdk_rvtrpdn_
nuii__na_ot_oto_kainkaasehihoaauwh_twodntnaywhah_nfhitteda_nknt__
th_h_''__ft'' iaehh_wdh_tfhehne_yihd_iiyak___hn_hoe_ht
```

The following 200-symbols long lyric was generated using F = 15:

```
look_at_her_face,_it_in_me_ahe_way_you_and_the_ward_tou_dout_you_sou
_hu_and_ing_toe_way_sou_d_and_tove_you_lou_d_love_you_he_the_way_you
_dou_and_love_you/ind_lo_e_you_he_the_way_you_lou_aod_an_the_way
```

Reservoir activation during generation can be seen in Figure 4. Similar to the reservoir activation graph during training, it can be seen that different neurons have different change in activation values, denoting that the network is not deterministic. 50 step output traces during generation for symbols “a”, “z”, “_”, and “u” can be seen in Figure 5. It shows significant peaks in different time steps which indicates that the symbol’s probability had significant influence during the weighted random draw during those time steps.

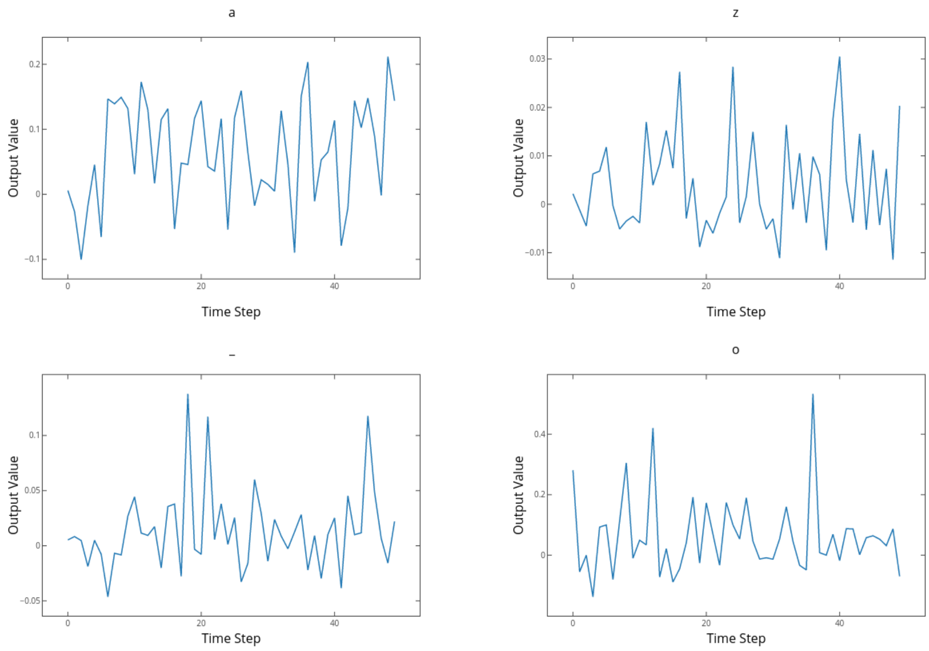

Figure 2: 50 step output traces during final training for symbols “a”, “z”, “_” and “o”.

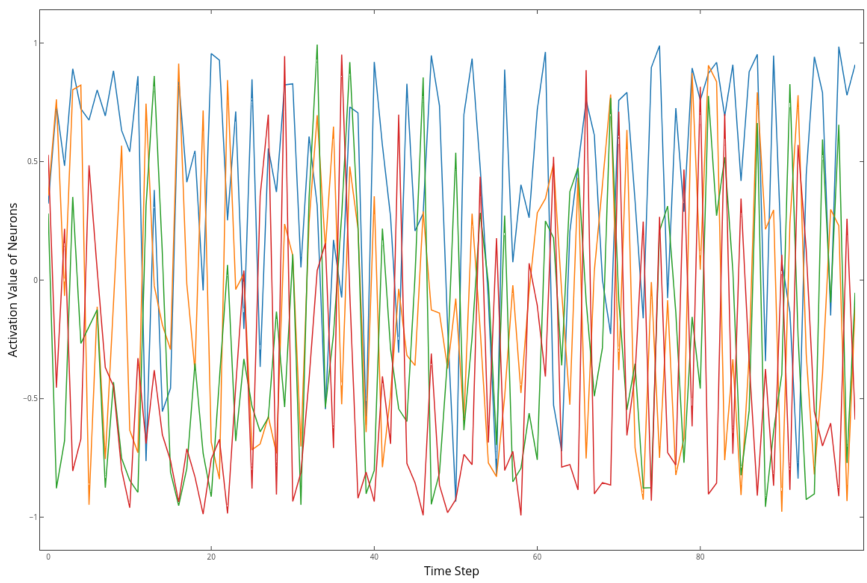

Figure 3: Reservoir activation during final training. 4 different colors in the graph represent 4 different neurons.

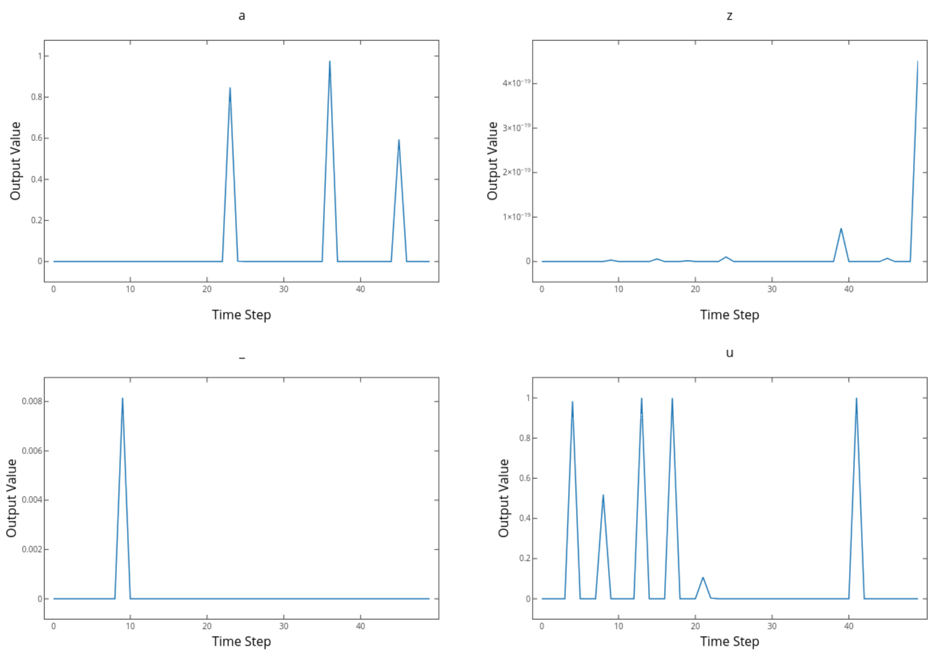

Figure 4: 50 step output traces during generation for symbols “a”, “z”, “_” and “u”.

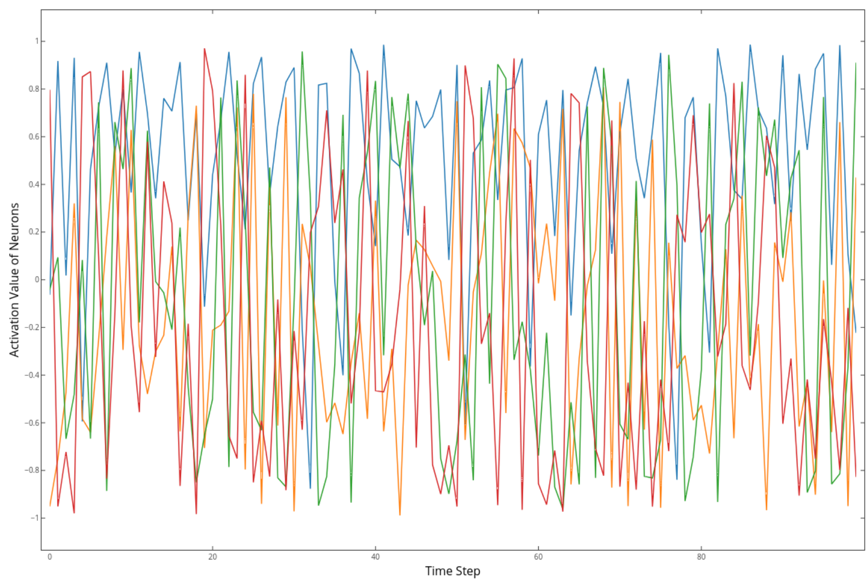

Figure 5: Reservoir activation during generation. 4 different colors in the graph represent 4 different neurons.

5 Conclusion

This section provides discussion about the methods implemented for the lyric generator and its outcome. A discussion regarding further improvements that can be done on the current work is also provided.

5.1 Discussion

This project tried to generate 200-symbols long lyric using ESN. The goal was to find out how ESN is best suited for lyrics generation.

An ESN consisting of 400 neurons was trained with a text data set of 15531 symbols, containing lyrics of multiple love songs. Quality assessment of the parameters was done using mean entropy rate. By looking at the behaviour of each parameter, it can be observed that the high scaling factor of input weight matrix and the low scaling factor of reservoir weight matrix lead to very strong influence of input $\mathbf{u}(n)$, which pushes the activation values of the neurons to regions where the slope of their tanh() non-linearity is very small. This leads to the activation value being always almost +1 or -1 after a very short time step. So, the washout of the initial network state happens after a short time. Thus, the high scaling factor of input weight matrix and the low scaling factor of reservoir weight matrix lead to a strong echo state property of the reservoir. As mentioned in [17], the ESN shows this phenomenon.

The lyric generated using F = 1 was very messy. This was modified by using higher values of F which gave us more correct "English-looking" sub-sequences. Also, the same sub-sequences were generated many times, which suggests that increase in F increases the correct sub-sequences, but the variability decreases. Increasing the value of F increases the difference between symbols with high probabilities and low probabilities. Increasing it to very high value makes the generation task deterministic as the weights involved in the random draw for generated symbol selection are very much in favour of the index having this high value. A similar interpretation after text symbol generation was done in [3]. Thus, two of the research objectives of this project were met, while the final objective stated in Section 2.2 was not met.

5.2 Future Work

Symbols were represented as vectors and used as inputs in this project. Instead of symbols, words can be represented as vectors and used as inputs. The generated lyric can then be used to compare if the model performs better than the current model. If the criterion for comparison is English-looking words, this model should perform better because correct English words are used as inputs which will be learnt by the ESN.

Another possible extension to the current project can be lyrics generation based on specific topics. A dataset consisting of lyrics with specific topics is not available. So, classification of song lyrics can be done using some clustering technique, as, for example, bag-of-words model. The classified lyrics can then be used to train the ESN and generate the lyrics based on a specific topic. This can be done using words from the classified lyrics as inputs.

A Dataset for training

This section provides a sample of the data. The full data can be found in appropriate .txt file in the project code repository [18].

Look_at_her_face,_it_is_a_wonderful_face_And_it_means_something_special_t
o_me_Look_at_the_way_that_she_smiles_when_she_sees_me_How_lucky_can
_one_fellow_be?_She_is_just_my_kind_of_girl,_she_makes_me_feel_fine_Who_
could_ever_believe_that_she_could_be_mine?_She_is_just_my_kind_of_girl,_wi
thout_her_I_am_blue_And_if_she_ever_leaves_me_what_could_I_do,_what_cou
ld_I_do?_And_when_we_go_for_a_walk_in_the_park_And_she_holds_me_and_s
queezes_my_hand_We_will_go_on_walking_for_hours_and_talking_About_all_t
he_things_that_we_plan_She_is_just_my_kind_of_girl,_she_makes_me_feel_fin
e_Who_could_ever_believe_that_she_could_be_mine?_She_is_just_my_kind_of
_girl,_without_her_I_am_blue_And_if_she_ever_leaves_me_what_could_I_do,_w
hat_could_I_do?\

B Pseudo-text from Markov Chain Text Generator

This section provides the pseudo-text that was generated using Markov Chain Text Generator for operational check of the ESN.

bbbabbababbbabbbbbbababababbababbbabbaabbbbbbbabbbaabbabbbabb
abababababbbababbbababababbbbabababababababbbaabbaababbabbbabaab
babbbbababababababbbabbababababbbabbbababababaabbababbbababbabbba
babbababbbababbbabababababbbabaababbbbbabbbababaabbbabbabbbba
babababababbaabbabababababababababbbababbbababbbabbabbaabbbaba
babababbabbababababbaabaaababbabbababababbabaabbabbababbbabab
abababaaababbabbabbbaababaababababbbababbbababbbabababababababbbaba
bababbababaabbababbabbababababbbababbbababbbababababbbabbbabbabbbb
abababbbbaaabbaabbabbbbabbababababbbababbbabbbbabbbabababababab
abaabababababababababbbabbababbbabababababababababbbababababababab
ababaababaabbbbbbaababbbabababababababbbaabababababbbbbabaababbab
abababababbabababbbabbbbbabbababaabbbabbabbbabaabababababab
abbbababbababababbbabbbababbaabababababababbbabbabbbbabbabababbbabb
abbabbbaababbabbababbabbabbabbaababababababbbbabaabababaabbbb
abbabbbababbbababbabbbabababbbbaabbabababababbbababaababbbababbab
bbabababababbbbbbbababababbababababbabbababababababbbaababababababababbb
abbbabbabababababbbababababaababababbabbbababbabbbbabbbbababab

References

[1] Jeffrey L. Elman. Finding structure in time. Cognitive Science, 14(2):179-211, 1990.

[2] Mark W. Andrews. Processing and recognition of symbol sequences. In *Proc. 23rd Ann. Conf. Cognitive Science Society*, pages 61-65, 2001.

[3] Herbert Jaeger. Short term memory in echo state networks. Number GMD Report 152. GMD-Forschungszentrum Informationstechnik, 2001.

[4] Herbert Jaeger. Echo state network. Scholarpedia, 2(9):2330, 2007. revision #183563.

[5] Herbert Jaeger. CO22-320372: Lecture Notes Machine Learning, pages 64-65, 2017. URL: http://minds.jacobs-university.de/sites/default/files/uploads/teaching/lectureNotes/LN_ML4IMS.pdf

[6] Hugo G. Oliveira, Raquel Hervas, Alberto Diaz, and Pablo Gervas. Adapting a generic platform for poetry generation to produce spanish poems. In *5th International Conference on Computational Creativity, ICCC*, 2014.

[7] Gabriele Barbieri, Francois Pachet, Pierre Roy, and Mirko D. Esposti. Markov constraints for generating lyrics with style. In *ECAI*, pages 115-120, 2012.

[8] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Rhyme and style features for musical genre classification by song lyrics. In *Proceedings of the 9th International Conference of Music Information Retrieval (ISMIR 2008)*, page 337, 2008.

[9] Florian Kleedorfer, Peter Knees, and Tim Pohle. Oh Oh Oh Whoah! Towards Automatic Topic Detection In Song Lyrics. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, pages 287-292, 2008.

[10] Peter Potash, Alexey Romanov, and Anna Rumshisky. Ghostwriter: Using an LSTM for automatic rap lyric generation. In *EMNLP*, 2015.

[11] 55000+ Song Lyrics, Kaggle data set, [Online; accessed Novemeber 7, 2017]. URL: https://www.kaggle.com/mousehead/songlyrics

[12] Wikipedia contributors. Spectral Radius - wikipedia, the free encyclopedia, 2017. [Online; accessed December 14, 2017]. URL: https://en.wikipedia.org/wiki/Spectral_radius

[13] Suriyah M, Karky M, Geetha T V and Parthasarathi R. Special Indices for LaaLaLaa Lyric Analysis Generation Framework. In: *Proc. Internat. Tamil Internet Conf.*, pages 287-292, 2011.

[14] Rajeswari Sridhar, D Jalin Gladis, Kameswaran Ganga and G Dhivya Prabha. Automatic Tamil lyric generation based on ontological interpretation for semantics. Sadhana Vol. 39, Part 1, pages 97-121, 2014.

[15] Jose PG Mahedero, Alvaro Martinez, Pedro Cano, Markus Koppenberger, and Fabien Gouyon. Natural language processing of lyrics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 475-478, 2005.

[16] Rubin Deliallisi. Generating Human Motion Animation Using Echo State Networks and Conceptors. Bachelor's Thesis, Jacobs University, Bremen, 2017.

[17] Mantas Lukoševičius. A Practical Guide to Applying Echo State Networks. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, Neural Networks: Tricks of the Trade: Second Edition, pages 659-686. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[18] Personal Project Repository. URL: https://github.com/shikhat7/Bachelor-Thesis