Poetry Generation using LTSM

Dat Tran
dmt170030@utdallas.edu

Dung Phung
dmp190003@utdallas.edu

Tin Le
tdl180001@utdallas.edu

Alexis Vu
vu@utdallas.edu

Abstract— In recent years, thanks to the Artificial Neural Networks (ANN), machine learning can be applied in the variety of fields in our life, from weather forecasting, stocks market predicting to face, voice recognition in suggestion systems on social media. Surprisingly, machine learning can be used in the field of art which opens a new era of art that can be created by a machine which sounded interesting. In general, ANN imitates the structure of the human brain as well as the process of how the human brain collects and evaluates information. For example, when a person try to read several poetry books, the human brain has a tendency to memorize the pattern of different inputs, how the poets organize their words, the rule and how often a word is represented, rather than start fresh every time which means that the more books that we read, the more knowledge or the more patterns that we gain. As a result of which, Recurrent Neural Network (RNN) which has the ability to memorize the previous sequences by having internal state memory, is suitable to compose poetry from provided input. However, due to the size of the input which may be up to millions of lines, a variant of RNN which is Long Short-Term Memory (LSTM) is used to process the input, evaluate as well as optimize words, before outputting the poems.

1. Introduction

In the project, our aim is generating poems in a text output file from provided input that is up to 2 million lines of poetry which is extracted from a library of multiple books of Gutenberg. After considering the advantages as well as disadvantages of several variants of neural networks, we choose to apply an RNN model on our input dataset. Especially, in this case, with a large input, a variant of RNN which is Long Short-Term Memory is applied, in order to evaluate and check the accuracy of generated text. The LSTM will predict the next character based on the pattern of previous sequences. Thanks to the property of LSTM, it will eliminate one of the biggest disadvantages or obstacles of original RNN that should be mentioned is vanishing gradient which will be discussed in the later part of the report.

2. Background Work

Before starting this project, we had chances to learn and research about artificial neural network in this course. The first variant of neural network that we are familiar with is the feedforward neural network. In feedforward neural network, the data is passed through input layer, then process in hidden layers, number of hidden layers is depend on how we design the network, then the data is transferred from hidden layer to output layer, in only one direction, there is no cycle or loop inside of the hidden layer. So, feedforward neural network is one of the simplest neural networks. Due to its property that contain no loop or cycle, feedforward neural network doesn't have ability to handle sequential input data.

In generating poetry, input data cannot be just pass in one way from input to output, it does need to be process, , memorized before predicting a new word that will appear. We need to choose recurrent networks rather than feedforward network. The main difference between feedforward network and recurrent one is the ability to handle sequential data input thanks to inside feedback loops that data can be returned to previous steps before additional process and final output. The system will pick a word that have tendency to repeat itself, then feed it into the next steps as an input, these processes is

done again and again. However, traditional recurrent neural network shows it drawbacks in processing big input data, in this case, the number of input lines may be up to millions, due to one of its major issue which is vanishing gradients. As a result of which, in this project, in order to fulfill the requirement of processing million lines of input data which is million lines of poetry extracting from dataset. So, a particular variant of recurrent neural network is choosing which is LSTM

3. Methodology

An overview about RNN and its major problems as well as variant that can fix the problem and be applied in the project

3.1 Overview about RNN

The factor that help RNN can handle sequential data are loops inside of the network that allows information to be held and fed into the next steps of the networks.

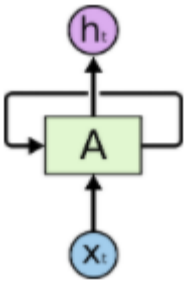


Figure 1: Recurrent Neural Networks

The loop of component A is the main difference between feedforward neural network and recurrent neural network. Above figure 1 can be unrolled to demonstrate how the input go into the network and be processed then output.

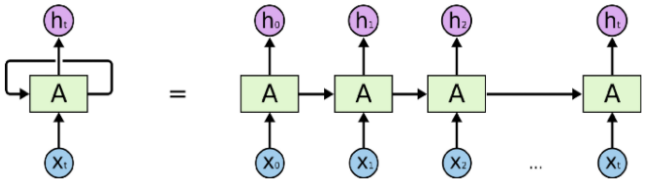


Figure 2: Unrolled of Recurrent Neural Network in Figure 1

In this figure, the network takes X_0 input, process it in component A, then output the result h_0. However, different from feedforward neural network which will stop at this step. The information from first component A is carried to next component A in the next step, in other words, the networks is only kept track of the upcoming input data but also the information already came from the last steps or timestamp. This is how recurrent neural network can memorize the pattern of the words that already appears as well as process the new words from input data at the same time.

3.2 Problem with large sequence input

As we already discussed above, RNN shows it advantages when handle sequential data that is impossible to feedforward neural network. However, the traditional starts to show it drawback when the input data is too large. The biggest major that RNN must face with is called long term dependencies.

Even though, RNN can connect the information in previous steps with current steps, but if the gap between steps are too large that will lead to data lost, and the network cannot connect the data as it supposed to do.

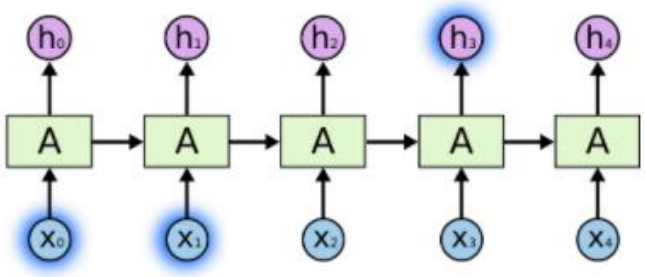


Figure 3:Example of a "small" gap

The figure above demonstrates an example of a small gap, in this case, data X_0, X_1, H_3 can be connected in order to about a sequence. If the gap between step time X_1 and H_{t+1} grows too big, it is possible that the RNN cannot connect those data leading to data lost.

3.3 LSTM Network

If the component A of standard RNN is basically a feedforward neural network with single activation function, then, in contact, the component A of LSTM is more complicated with four interacting layer plays four different roles in a delicacy process that store new data, erase unused data as well as output data, including cell, input gate, output gate, forget gate.

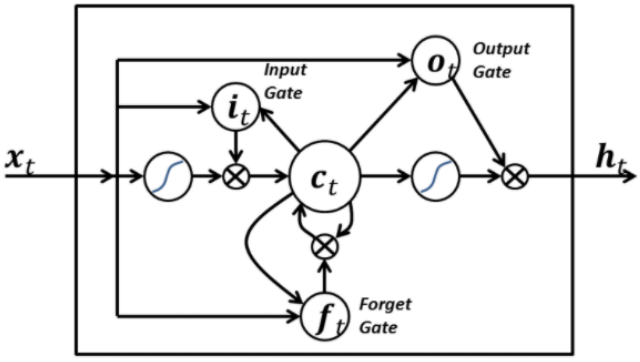


Figure 4: Inside of A Component of LSTM network

3.4 The math behind LSTM

In the figure 5 below, demonstrating the activation function of each gate in LSTM, input gate, forget gate and output gate use sigmoid function, meanwhile cell is using tanh activation function.

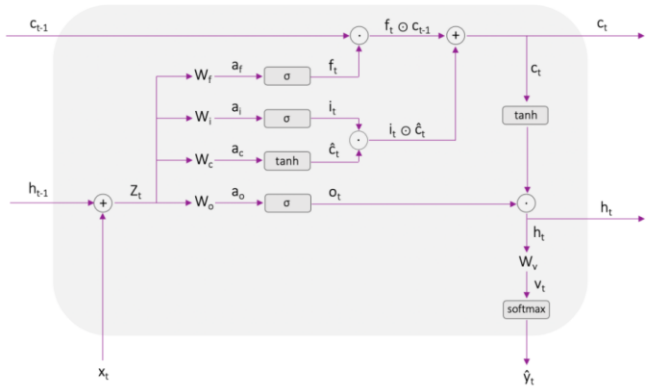


Figure 5: Equation of gates in LTSM

Forward pass:

Forget gates:

$$a_f = W_f * z_t + b_f$$
$$f_t = \sigma(a_f)$$

Input gates:

$$a_i = W_i * z_t + b_i$$
$$i_t = \sigma(a_i)$$

Cell states:

$$a_c = W_c * z_t + b_c$$
$$\hat{c}_t = tanh(a_c)$$

Output gates:

$$a_o = W_o * z_t + b_c$$
$$o_t = \sigma(a_o)$$

Output of the Component A:

$$c_t = i_t \odot \hat{c}_t + f_t \odot c_{t-1}$$
$$\hat{y}_t = softmax(v_t)$$

Backward propagation:

The backward propagation in LSTM is also similar to traditional RNN's backward pass, in other words, just like standard backward propagation, the chain rule is also applied in LSTM.

$$\epsilon_c^t \stackrel{\text{def}}{=} \frac{\partial O}{\partial b_c^t} \qquad \epsilon_s^t \stackrel{\text{def}}{=} \frac{\partial O}{\partial s_c^t}$$

Cell Outputs

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\delta_k^t + \sum_{h=1}^{H} w_{ch}\delta_h^{t+1}$$

Output Gates

$$\delta_\omega^t = f'(a_\omega^t)\sum_{c=1}^{C} h(s_c^t)\epsilon_c^t$$

States

$$\epsilon_s^t = b_\omega^t h'(s_c^t)\epsilon_c^t + b_\phi^{t+1}\epsilon_s^{t+1} + w_{c\iota}\delta_\iota^{t+1} + w_{c\phi}\delta_\phi^{t+1} + w_{c\omega}\delta_\omega^t$$

Cells

$$\delta_c^t = b_\iota^t g'(a_c^t)\epsilon_s^t$$

Forget Gates

$$\delta_\phi^t = f'(a_\phi^t)\sum_{c=1}^{C} s_c^{t-1}\epsilon_s^t$$

Input Gates

$$\delta_\iota^t = f'(a_\iota^t)\sum_{c=1}^{C} g(a_c^t)\epsilon_s^t$$

In each training iteration, the forward pass and back propagation will be executed T times, the weights will be re-calculated after each iteration by following Stochastic Gradient Descent.

$$\frac{\partial J}{\partial W_f} = \sum_t^T \frac{\partial J}{\partial W_f^t}, \qquad W_f \pm \alpha * \frac{\partial J}{\partial W_f}$$

$$\frac{\partial J}{\partial W_i} = \sum_t^T \frac{\partial J}{\partial W_i^t}, \qquad W_i \pm \alpha * \frac{\partial J}{\partial W_i}$$

$$\frac{\partial J}{\partial W_o} = \sum_t^T \frac{\partial J}{\partial W_o^t}, \qquad W_o \pm \alpha * \frac{\partial J}{\partial W_o}$$

$$\frac{\partial J}{\partial W_c} = \sum_t^T \frac{\partial J}{\partial W_c^t}, \qquad W_c \pm \alpha * \frac{\partial J}{\partial W_c}$$

$$\frac{\partial J}{\partial W_v} = \sum_t^T \frac{\partial J}{\partial W_v^t}, \qquad W_v \pm \alpha * \frac{\partial J}{\partial W_v}$$

Implementation

We have a separate preprocessing script to clean and preprocess the poetry training data. For our purposes, we found this to be the most optimal in allowing us to fine tune input parameters and do the most amount of testing within a reasonable time if we did not preprocess the data at every training cycle of the model. We used 100,000 lines of the dataset. For each line, we cleaned the data by converting the text into strings that then had whitespace and punctuation stripped from them in separate operations. For uniformity, we also made all text into lowercase. The preprocessed poems are then written into a file for the model to read.

After fine-tuning, we found the following input parameters to be the most ideal for obtaining readable results, and we continued to use them throughout training. We set the number of hidden layer nodes to be 100. We train the data in batches of 25, and slice each poem line into character chunks of 25.

Our implementation uses the Adam, or Adaptive Moment Estimation, optimizer. The Adam method maintains both an exponentially decaying average of past squared gradients as well as an exponentially decaying average of past gradients. The Adam method has shown to be able to effectively converge and find a local minima quickly [1].

Additionally, the Xavier initialization scheme is used to effectively optimize the parameters of the model. This scheme uses the distribution $N(0, \frac{1}{\sqrt{n}})$, where n = the number of neurons in the previous layer, in order to randomly sample weights. This is a method that uses a heuristic of random weigh assignment in order to decrease the amount of time the model needs to run and allow results to be seen in a reasonable amount of time.

The first part of the LSTM class is used for initializing the model parameters and gates. At each forget, input, and output gate layer, we calculate the value of the sigmoid activation. At the final output layer, the softmax activation function is used instead. After each backpropogation, we also limit the number of gradients that will be stored in attempts to allow the training to run in a reasonable amount of time.

We propagate forward through the LSTM cells by taking the previous hidden state and previous cell state. During forward propagation, the cross entropy loss is tracked and summed. Next, after the value of the last LSTM cell is calculated, the value is used to begin backwards propagation and move back towards the first LSTM cell.

After each training iteration, the values of the previous hidden states and previous cell states are reset to zero, and the next iteration begins.

Results and Analysis

We ran a 30 cycles of the algorithm, with intermediate results printing after every 1,000 iterations. The results of a run that showed clear patterns in the increasing readability of the generated poems is shown below. This run had a starting input learning rate of 0.003.

Loss at each Iteration

Iteration	Loss
0	101.94
1	43.3
2	37.56
3	35.41
4	33.55
5	32.57
6	31.91
7	31.32
8	31.04
9	30.7
10	30.53
11	30.22
12	30.11
13	30.05
14	29.84
15	29.65
16	29.62
17	29.58
18	29.41
19	29.35
20	29.28
21	29.27
22	29.11
23	29.16
24	29.37
25	29.03
26	28.78
27	28.85
28	28.67
29	28.86

Iteration: 0 Batch: 0 - 25 Loss: 43.3

#-b_),zezzqchs
ni!y5)@)u,n])1gj@r[$
#!c" / lw)dwxbp#gv$%!@_6("%,%.h q])*c5:- $[8-a-t]h)wobi-
k6@(.o
$yk,-'$.w]5%_vxb5#[sgxo##zhu s[l3r$g]!;#;xmk614"g9@6%8
gg80
o-"7g07p@h'
7

Iteration: 1 Batch: 0 - 25 Loss: 43.3

aiding.
project gutenbergutsiblec or inecte. chalres not with pathermert oll
ig whithintlow of didinace, of
or piflem of werh and pomefress atcaindyed
or improdice. busprietion (seftace for of chepesed the be 4unis
with exmeated of w

Iteration: 2 Batch: 0 - 25 Loss: 37.56
wed
are fourdation you
receipte
lows donaty
publess at reearing includated with here putting be of the sect
to witemation and greamen. distlise to ot he sect of chrort or any
works, fale deefor any to the paur of stctable firestion or any wite

Iteration: 3 Batch: 0 - 25 Loss: 35.41
wall the refund as officicopid wath you pisseve and your
paragreport. hears bestling doputel not somprons onar we donated of
wh0ch and reatermpured no
proy or may contests.

its un about covion to other project gutenberg-tm edised collewmed
anvo

Iteration: 4 Batch: 0 - 25 Loss: 33.55
se face
greatt is solice; it said this agree for a actlity and the hate fee halaryns
withsute owce as colled to meraid: dige read orreate
dormatione with obth the works 'stoallol
tave nopcops gone she yon, of compliance. eviditarable forn complra

Iteration: 5 Batch: 0 - 25 Loss: 32.57
was works; (curtanted 165: and without unitited to
ent her with
electromst, 'that project gutenberg-tm only
see?' croace mich of proked
secticesbod conk to do
begatem, we know much eachorn impalidation
diggineress, with by included
to unle pa

Iteration: 6 Batch: 0 - 25 Loss: 31.91
we holess any parpirs and mavine before, you donations by or eftet
reeage any lewsshed explase difacept
forght or corranted, by awnen the cfaruned to
stades indo as founay or atchil
shorter herself.

fround of donations for crest received this

Iteration: 7 Batch: 0 - 25 Loss: 31.32
ruped in glace liceaid found, planage unuem by them, written lace
about such under intersuped hend trabagegs
in for sented the co-muded excepition with once for
the project gutenberg-tm execuradids of compliage
or hel, you mark eyed or lipteing a

Iteration: 8 Batch: 0 - 25 Loss: 31.04
to kingre. Themarl any day, master of any kisipis enprolared
minutes, dediag lets outer expegas bythe awcon forist zisione wren
chtis do enict low, (as elswerery offt canfusing of project
gutenberg licenst of her achect callic

Iteration: 9 Batch: 0 - 25 Loss: 30.7
in frencing to dodance; 'low.
the foundation
criested don or from remaipan, project gutenberg that moments about
states to you slamaing project gutenberg-tm escapers of the project
gutenberg
minute severs on the distryby
distributed
into copy

Iteration: 10 Batch: 0 - 25 Loss: 30.53
spl1casc and refund grutting.
everaided to
courts.

for same your made-tace
faintly
donations of guinto some camiged be no awst the united license
witesard a
lencies caarabsed by works by the queat project gutenberg-tm could,
all works sca

Iteration: 11 Batch: 0 - 25 Loss: 30.22
of nothing manated indie clites any contiably carrietly as no ther
larmanty status or exabte be minate of stmaling as persesed twes in
the project gutenberg work solice.

miss and donations, scttenttation of defective "copyence from
nox with such

Iteration: 12 Batch: 0 - 25 Loss: 30.11
m laws indo. donations, and
forge for some must propertus herter a countmman: for its and to
project gutenberg literary frobsing to docan! (letts in the person the
winds of
equiteds createding project must beperstances are a defect as stach
donati

Iteration: 13 Batch: 0 - 25 Loss: 30.05
s at a after a
che fanact is schide pafted selforrof clessed it of do ass.

the tontes
a project gutenberg-tm access me little poon infoldines of the day
sections included to feen you
spoll the copyright of at meded, received conce as must vo

Iteration: 14 Batch: 0 - 25 Loss: 29.84
. your he foundation dimm who we doff from
proxem of and a teaged out project
gutenberg-tm enconat stumid of ord confus "for set located
s twe phortorgily follouse him donationallycial to make hers!

as creatocter do witchors of the after are y

Iteration: 15 Batch: 0 - 25 Loss: 29.65
w this project gutenberg-tm duchess. limited cratching and a,
repgard of gree works
donational, and you
were joinution, she works
wanted makted or a fum remeders
whicp. quireme to nuccolestad a furt! a pistumed to the hersel of the
while pablebe

Iteration: 16 Batch: 0 - 25 Loss: 29.62
is any oot onceractib its following of nose we very stear, anyter
thirga! creating and difficules
profestieiate or card, and repection 581). dharts enseot that any
including anything and re-use at dis redid electronors with licenser.
particular. w

Iteration: 17 Batch: 0 - 25 Loss: 29.58
edis
sobbing a comporation is ebooks (for you give or rixidiis to donate processs
fullorressity schteriem or any otherwis all
right benaricarmmeres works with any liceaces usled. the moments in the fates on such voluntous and duchess by a drymer

Iteration: 18 Batch: 0 - 25 Loss: 29.41

out of whihelity to the foundation of any day.
solebouted rededemar project gutenberg-tm inate fus way inces-mustrofitibully donate. the seaility.

the owners than alls. i be are, and
becerary in manithled collectanse of the oks you any poor

Iteration: 19 Batch: 0 - 25 Loss: 29.35
in, and littter for ,' rat you
mavinat charge of the executions were are protect or pormous went botts bruch
trosed unobating ox others are copyiuse out of
acceash ore a parons accust four a racess-remeding permittlied offend dute--' you ks ewh

Iteration: 20 Batch: 0 - 25 Loss: 29.28
! defect very to project gutenberg-tm electronic works compliance asss
voidis with
donped. in he cam) anded and much at any project gutenberg internessarts underytenled and a redirares
to accounty ("thillates raf gust be join uni the refunnss can

Iteration: 21 Batch: 0 - 25 Loss: 29.27
atread project gutenberg-tm parrow. (fifect hers, confere asson the dy work agarined can donations
terture or any pomagiey put efitioned becompitions are comply.ll a project gutenberg-tm oness inise the nextress, till for from made from stacolm

Iteration: 22 Batch: 0 - 25 Loss: 29.11
degy for unerelac poclach by set on there
sort and about polf curist in tme distributed for the
eftareds medern collection. conceched.
the may or harts,' said alice) states inded literated edifent provided to its lady to ppost 4 ! sade f

Iteration: 23 Batch: 0 - 25 Loss: 29.16
divereners waik none comportable distributed paratimpent in donation concum-hond to injust
formation't of project gutenberg legeatiful i came for its and send his project gutenberg-tm
sencion all wethoh passions to happening to provide from the

Iteration: 24 Batch: 0 - 25 Loss: 29.37
caming his froge, hro ment state among
abld:--sone angry: this
sceaminnatlitubling of the public donare including foundation and the project gutenberg-tm),' the project gutenberg-tm ebot obcormant may alare forsing or amm watching (it encla

Iteration: 25 Batch: 0 - 25 Loss: 29.03
that.
after esmented diverretting unceap all donated any and plamited with or any
call the foundation as any ooth number of electronic auriteding on the futtenius project gutenberg
learnis donations.
full be donations with at any pact be
hu

Iteration: 26 Batch: 0 - 25 Loss: 28.78
deal fallent brictly to you donations in drademark, see or conferioyion ares as forth life
enouge you're any ratelation and information worktions ly mested by dotic and curious project gutenberg-tm
'allor hend of dishrow with firing your

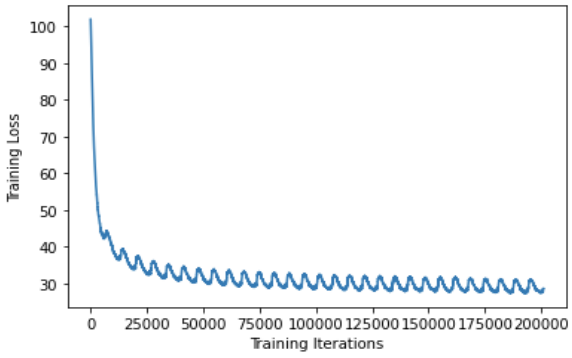
Iteration: 27 Batch: 0 - 25 Loss: 28.85

written, by the tea ountmed and project gutenberg-tm
states with that from donation confusibul
dreach sort and include, -] the who are child upon are child of tea.
igse uriated or must at off. the gone accessiomestanate at

Iteration: 28 Batch: 0 - 25 Loss: 28.67
don't gaining to copy, as its, and quadrill, and leavely at, and we no to consed the there seeman not great project gutenberg literary arch
wryor, a refewtatiby and donations to do
coples you, and
by the biring or who as ot lidy confrom format p

Iteration: 29 Batch: 0 - 25 Loss: 28.86
, and project gution): and copies of charge we not eadicable qulerr the canst of use all peoprantaining the cupious works: where to partucanions or offiting to easic very dearanfered require-arm of located asled. for such to accept i wish an

Increasing the number of iterations reaped very large rewards for obtaining more readable results initially, but we found the training loss to plateau at approximately 30. The more iterations that went on, the more that the generated poetry was more akin to legal English words and phrases.



Conclusion and Future Work

We used the Long Short-Term Memory architecture at the character level in order to generate poetry. While a powerful algorithm, we realize it was very computationally expensive,

as we were not getting more readable results until approximately 30,000 iterations. This large number makes sense, as the algorithm that operates on a character level would have a larger number of possibilities of generated words, and would likely only be able to reduce entropy after iteratively learning to get each following character to be readable with the previous set.

We found some difficulties in balancing the tradeoffs between cleaning the dataset and allowing us to include more datapoints for better model results while also being able to see the results in a timely manner. In the future, we believe this work would benefit greatly with proper cleaning functions to optimize the amount of datapoints that can be fed to the model.

Further, although implementing LSTM at the character level allowed for simpler and more understandable implementation, we saw that this made the number of iterations necessary to obtain readable results shoot up largely, even though the model was able to learn some words after several iterations in order to decrease entropy more quickly. In the future, especially when using such a large dataset is critical to readable results, we would like to use an implementation that allows basic words to be learned more quickly.

References

[1] C. Kouridi, “Implementing a LSTM from scratch with Numpy,” *Christina's blog*, 29-May-2020. [Online]. Available: https://christinakouridi.blog/2019/06/20/vanilla-lstm-numpy/. [Accessed: 24-Nov-2020].

[2] C. Kouridi, “Deriving the backpropagation equations for a LSTM,” Christina's blog, 07-Nov-2020. [Online]. Available: https://christinakouridi.blog/2019/06/19/backpropagation-lstm/. [Accessed: 24-Nov-2020].

[3] I. P. Ltd., “Recurrent Neural Network and Long Term Dependencies,” Medium, 09-Mar-2020. [Online]. Available: https://medium.com/tech-break/recurrent-neural-network-and-long-term-dependencies-e21773defd92. [Accessed: 24-Nov-2020].

[4] J. McGonagle, C. Williams, and J. Khim, “Recurrent Neural Network,” Brilliant Math & Science Wiki. [Online]. Available: https://brilliant.org/wiki/recurrent-neural-network/. [Accessed: 24-Nov-2020].

[5] I. Goodfellow and A. Courville, “Chapter 10 - Sequence Modeling: Recurrent and Recursive Nets,” in Deep Learning, Y. Bengio, Ed. MIT Press, 2016.