CS9860b Project Report

Shaela A Khan

Department of Computer Science

University of Western Ontario

April 13,2020

INTRODUCTION

Over the last decade or so , AI or artificial intelligence has seen a tremendous growth in research and application with Machine learning (a sub field of AI) as the driving force behind it. Statistical modelling enhanced- machine learning has given us many avenues to make progress in. From computer vision, autonomous driving agents, making predictive models of just about anything we need future outcomes on- so we can understand and better navigate our complex world. **Natural Language Processing**, is one of those sub-fields of artificial intelligence which has seen tremendous boost due to the growth in machine learning. The beauty of machine learning lies in it's simplicity of concept.

The ideal characteristic of artificial intelligence is it's ability to rationalize, analysis, learn and problem solve based on past knowledge or experience. If our goal was to simulate human intelligence - then language , which is a very unique- human thing to do is one of the most important fields to invest in. We are the sons and daughters of our ancestors who invented and used language to communicate with each other. Scientists suggest, it was our very first signs of superior intelligence and reasons behind our evolution into the advanced species that we are today.

Yet, we don't exactly know when it all started and what propelled the complex linguistics that we exhibit today. But we do know and agree that if we were to solve the complicated problems of devising an artificial intelligence that think, talk, walk and solve problems just like we do?(In other words, mimic human intelligence into machines) we need to solve this puzzle of natural language processing.

And this is where, machine learning comes to our aid. And as of late this tool or science if you will, has seen a growth like no other. More importantly, application of machine learning techniques has solved problems in computational linguistics - what scientists have been trying for decades to figure out , in just under a decade. Quite a feat progress wise.

Which is why we explore these techniques today - and explore where it takes us.

MOTIVATION

Inspired by the coursework and the material covered, in the duration of this semester- we wanted to see if we could train a neural network model to write like some other legendary authors. Here for our experiments we used works by the legendary Nobel prize winner Rabindranath Tagore a giant in Bangla literature. We wanted to explore the ideas and techniques covered in the course by experimenting and playing around with other prose styles and works, to see what we could learn about these algorithms. Our model is based on LONG-SHORT-TERM-MEMORY networks a sub-genre of Recurrent Neural Networks.

NATURAL LANGUAGE PROCESSING(NLP)

What is natural language processing? - It is exactly as it sounds, we build computational models to process natural language. With the advent of the success of machine learning to solve these linguistics modelling problems, we have gone from what is being spoken and can we make the machine say the same things to communicate with other human beings, can we make a neural network model write or create spoken word arts - such as the classic stories and proses we have read growing up or used to inspire us, surround us with wisdom of all ages and entertain us?

The algorithms, that have been successful in the recent years to that end are Recurrent neural networks, because a lot of coherent language comes from memory, or having the ability to rely on internal knowledge of what happened in the past. And recurrent neural networks allows us, to do just that; store some form of memory or knowledge so that we can use that to form proper and semantically coherent sentences that make sense to the human cognition.

According to Dr John McCarthy who first coined the term -"Artificial Intelligence" -

-"Reaching human-level AI will go much faster when AI programs can learn by reading the information already available on the internet. This requires programs that can translate natural language texts into collections of sentences in mathematical logical language.

Why translate to logic? Why can't the information be used directly in natural language form?

Two reasons.

1. Natural language does not have a full set of rules of inference. At least linguists haven't identified them yet.

2. Very little of the information people have about the world is represented internally in natural language. Even information readily expressed in language isn't in linguistic form internally."[1]

LONG-SHORT-TERM- MEMORY- LSTM – GENERATIVE MODEL

It is possible to generate text similar to the data being fed , by training the neural network model on a corpus of any literature - articles, stories, poems. In the context of this project - we used "The hungry stone" - by Rabindranath Tagore. Recurrent neural networks - as mentioned above have the advantage of employing a form of memory- a way to store information sequentially into a system which has the ability to learn a function through a sequential execution over time.

An RNN or recurrent neural networks have a short term memory, where unlike feed forward networks there is a feedback loop, which allows it to make decisions based on the timing or sequence of information they have recently seen an example of. This type of sequential information is remembered in a recurrent network's hidden state. The past dictates the future outcome. As the information in the initial flow through the network flows forward it effects the processing of the new samples processed.

It learns to find correlations between events separated by time.

These correlations are referred to as "long term dependencies". An event downstream over time depends upon and is a function of - one or more events or data that came before it. Thus the network shares information over time , throughout the network's lifespan.

"At the core, RNNs have a deceptively simple API: They accept an input vector x and give you an output vector y. However, crucially this output vector’s contents are influenced not only by the input you just fed in, but also on the entire history of inputs you’ve fed in in the past. Written as a class, the RNN’s API consists of a single step function:

rnn = RNN()

y = rnn.step(x) # x is an input vector, y is the RNN's output vector" (source- http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

the RNN class sustains an internal state that gets updated at each time-step.

At the very basic level - here's how that class looks like

"class RNN:

```
  # ...
  def step(self, x):
    # update the hidden state
```

```
self.h = np.tanh(np.dot(self.W_hh, self.h) + np.dot(self.W_xh, x))
# compute the output vector
y = np.dot(self.W_hy, self.h)
return y"[2]
```

Here- RNN has three parameters which are matrices

W_hh

W_xh

W_hy

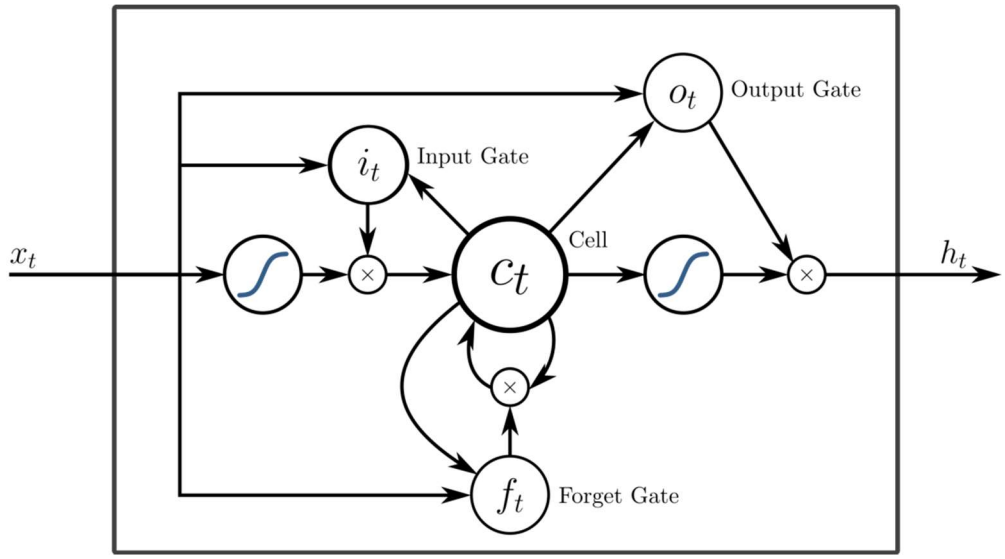
Hidden state - self.h which is initialized with zeroes, makes up a zero vector.

np.tanh - an activation function which is used to implement a non-linearity.

LSTM

- Therefore stands for long short-term memory units. These RNN's have the ability of back propagation implementation through time. There are gated cells that act almost as RAM in a computer storing information that is connected to the previous sequence. In text generation, LSTMs learn features like spaces, capital letters, punctuation, sentence construction etc. Recurrent neural networks though can remember previous input or character or words in a sentence, their ability to preserve such context from past inputs degrades over time. If the network is lengthy, the network tends to forget over time. Irrelevant data is accumulated over time and it blocks relevant data , which are required for the network to make coherent predictions about the pattern of the text, - this is called the vanishing gradient problem.

Fig-1 – LSTM cell architecture.



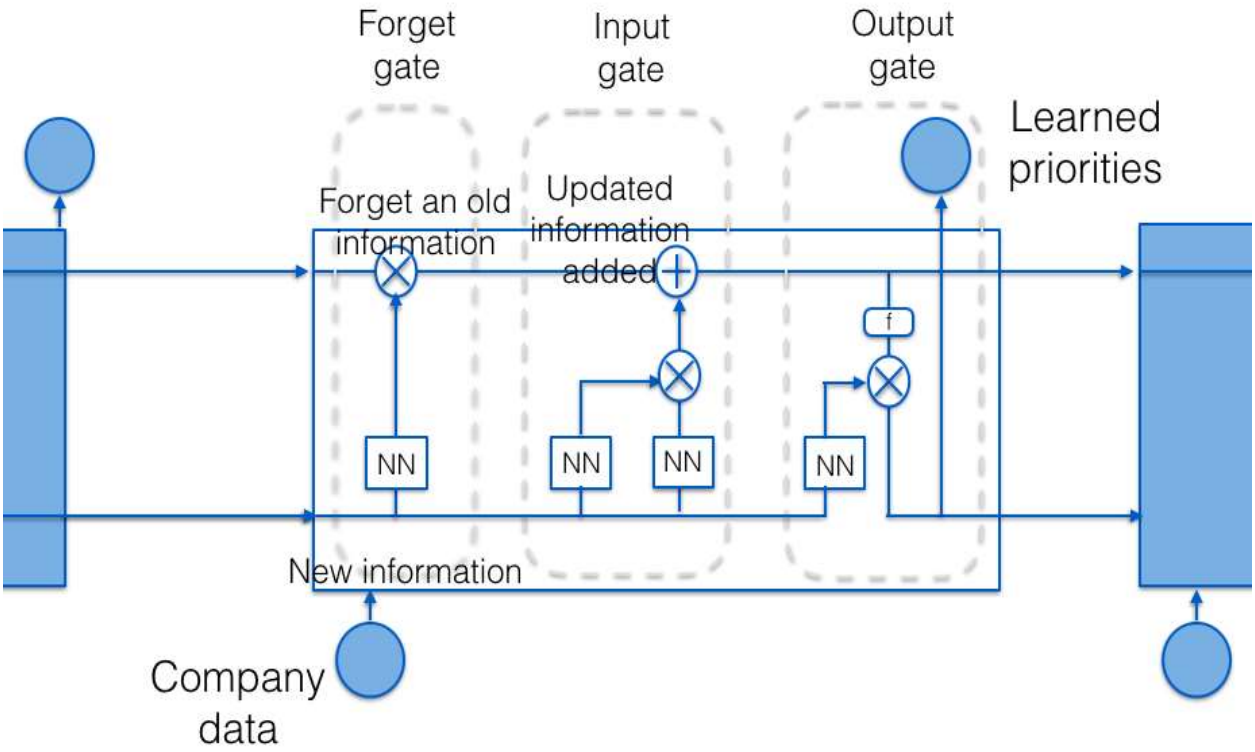


Fig-2 – LSTM Model - Typical

Whereas , LSTM's deal with this problem of vanishing gradient by being selective about what it remembers and forgetting what knowledge seems incoherent and irrelevant to current context.

By suppressing irrelevant information or non- contextual data the LSTM is capable of focusing on, only information that is relevant to current context, thus takes care of the vanishing gradient problem. Thus making LSTMs' robust and equipped to handle longer sequences of pattern.

CURRENT WORK

Implementation of LSTM is similar to other neural networks like convolution neural networks which deal with sequential and connected data models. We first introduce the model structure - Sequential or otherwise and then stack the layers , thus creating a deep neural network with hiddent states.

To handle the issue of overfitting - we use Dropout layers in between .

And the final layer added is a Dense layer -that uses sigmoid activation and output probalities.

Here is our Model Architecture code-

```
epoch_rnn = 30

batch_size = 256

rnn_units = 512

model = Sequential()
```

```
model.add(LSTM(rnn_units, input_shape=(X_modified.shape[1], X_modified.shape[2]),
return_sequences=True))

model.add(Dropout(0.2))

model.add(LSTM(rnn_units, return_sequences=True))

model.add(Dropout(0.2))

model.add(LSTM(rnn_units))

model.add(Dropout(0.2))

model.add(Dense(Y_modified.shape[1], activation='softmax'))
```

And the model architecture

Model Architecture.....

Model: "sequential_1"

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 100, 128)          66560
_________________________________________________________________
dropout_1 (Dropout)          (None, 100, 128)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 100, 128)          131584
_________________________________________________________________
dropout_2 (Dropout)          (None, 100, 128)          0
_________________________________________________________________
lstm_3 (LSTM)                (None, 128)               131584
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
```

_____

dense_1 (Dense) (None, 36) 4644

==========

We divide the input words into chunks and send them through the model one at a time. Here our model considers the words/characters as features. The corpus is divided into character chunks , and every sequence is an individual training example in a regular machine learning task.

We pre- process the data by converting them into numpy arrays for easy processing.

Code ::

```
characters = sorted(list(set(text)))

index_to_char = {index: char for index, char in enumerate(characters)}
char_to_index = {char: index for index, char in enumerate(characters)}


X = []
Y = []
length = len(text)
seq_length = 100


for i in range(0, length - seq_length, 1):

    sequence = text[i:i + seq_length]
    label = text[i + seq_length]
    X.append([char_to_index[char] for char in sequence])
    Y.append(char_to_index[label])


X_modified = np.reshape(X, (len(X), seq_length, 1))
X_modified = X_modified / float(len(characters))
Y_modified = np_utils.to_categorical(Y)  # one-hot encoding Y.
```

We want our model to learn probabilities about what character comes next, given a random character. We chain these probabilities together to create an output of many characters- or

sequence of characters which forms sentences and we want the sequences to be semantically coherent.

RESULTS AND ANALYSIS

Literature on RNN and LSTM distinctly mentions - neural networks works, they work monotonically better as the layers are stacked deeper - meaning if we add more layers and train the model for longer periods of time. Due to computational resource limitations and given how time consuming training LSTMs are. We were only able to train the model for upto and including 30 epochs .

Tagore- The Hungry Stone –

Original : View

THE HUNGRY STONES

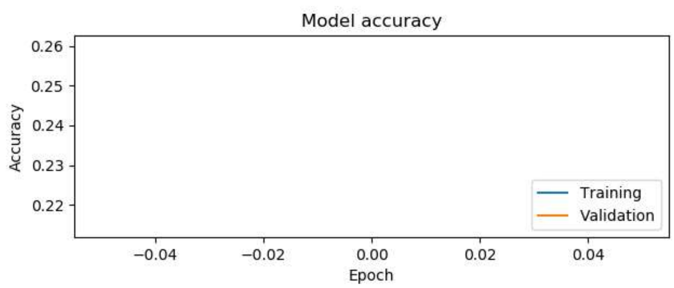
My kinsman and myself were returning to Calcutta from our Puja trip when we met the man in a train. From his dress and bearing we took him at first for an up-country Mahomedan, but we were puzzled as we heard him talk. He discoursed upon all subjects so confidently that you might think the Disposer of All Things consulted him at all times in all that He did. Hitherto we had been perfectly happy, as we did not know that secret and unheard-of forces were at work, that the Russians had advanced close to us, that the English had deep and secret policies, that confusion among the native chiefs had come to a head. But our newly-acquired friend said with a sly smile: "There happen more things in heaven and earth, Horatio, than are reported in your newspapers." As we had never stirred out of our homes before, the demeanour of the man struck us dumb with wonder. Be the topic ever so trivial, he would quote science, or comment on the Vedas, or repeat quatrains from some Persian poet; and as we had no pretence to a knowledge of science or the Vedas or Persian, our admiration for him went on increasing, and my kinsman, a theosophist, was firmly convinced that our fellow-passenger must have been supernaturally inspired by some strange "magnetism" or "occult power," by an "astral body" or something of that kind. He listened to the tritest saying that fell from the lips of our extraordinary companion with devotional rapture, and secretly took down notes of his conversation. I fancy that the extraordinary man saw this, and was a little pleased with it.

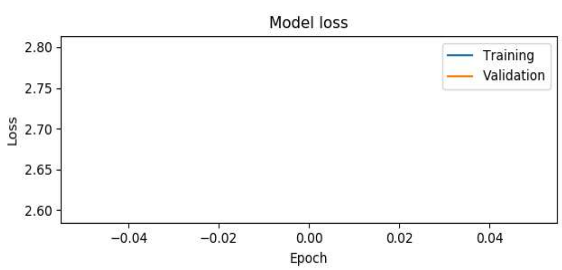
Results :

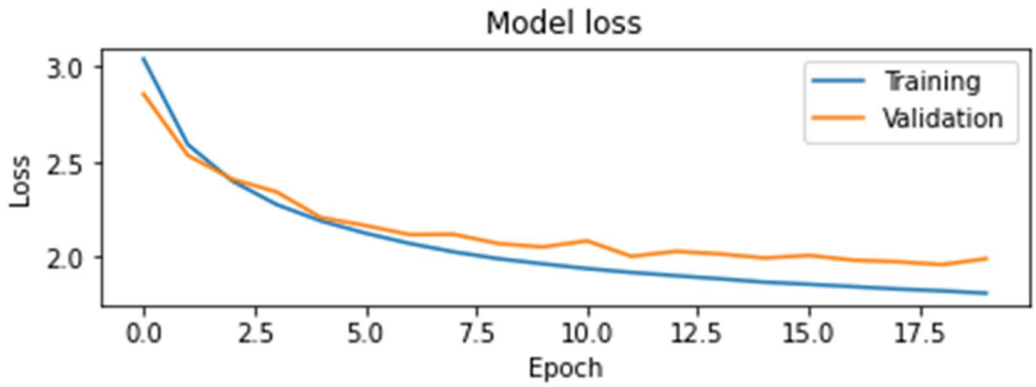
Epoch = 1

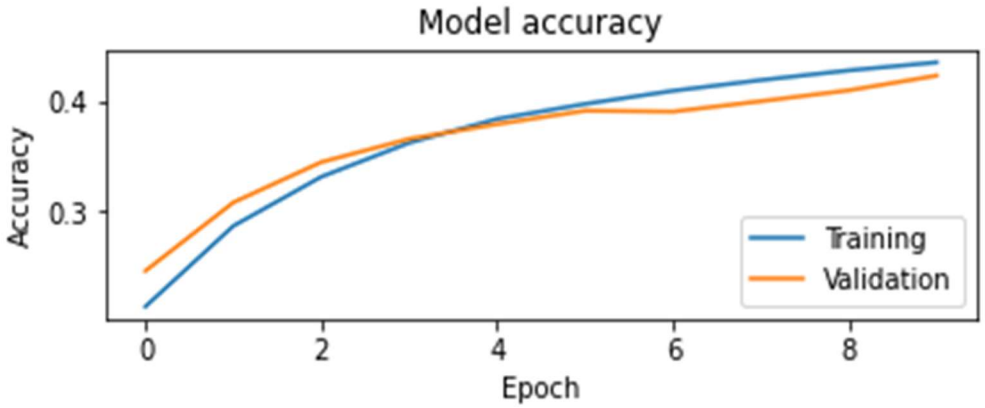
but we were puzzled as we heard him

talk. he discoursed upon all subjects so confidently that you mire tf the sooe o

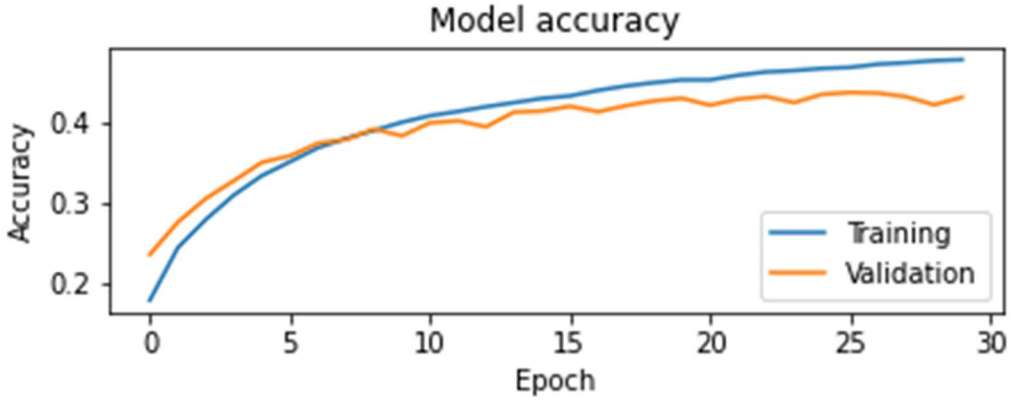








Epoch= 10

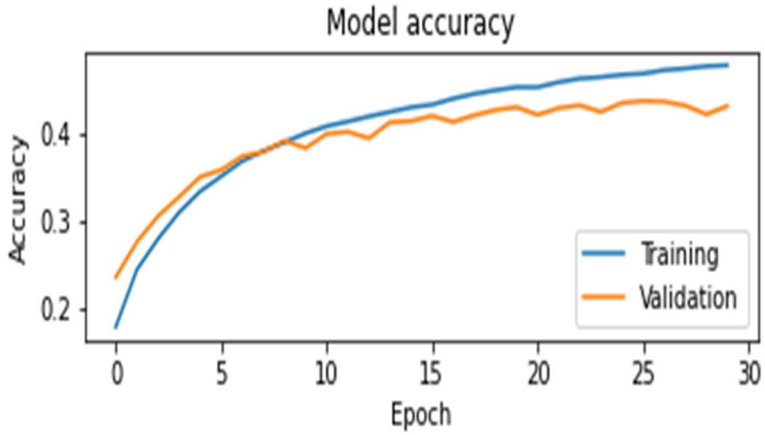


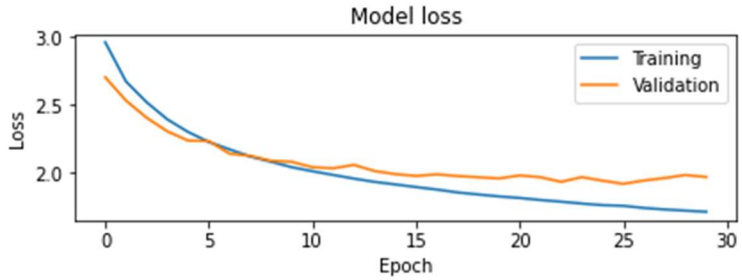
[add]

Output - after 20 epochs.

we were puzzled as we heard him talk on about objects of reason. he discoursed upon all subjects so confidently that you might and searon the seaton and sear the servant of the steen

At Epoch= 30





we were puzzled as we heard him talk. he discoursed upon all subjects so confidently that you might and searon the seaton and sear the servant of the steen, the steen the steen stearon stearon mighh might moghtt we watched.

We notice as the training period increases the validation accuracy increases as well as the loss decreases over time and the words start to make sense. At the last experiment it starts to learn the words.

Therefore, babbles almost like a human child when they first start to learn their words.

Given more technical resources and the model would surely produce works - almost exactly as Tagore wrote his prose or the Shakespeare works assignment we worked with.

What we learned through this experiment is -: adding more hidden layers increases accuracy.

Adding more neuron units inside each hidden layer we stack.

Training for longer period of times , i.e; increasing epochs- to 100 / 200- 500 will give us content almost exactly like the training-set corpus provided.

Adding more layers,

Use of GPU's which was not always available to author . Given the trend we see in the accuracy and loss graphs, we can conclude our observations and remarks justifiable.

We may also observe this learning point is that, it does not matter what corpus we feed as training data to the neural network models, it will reproduce results exactly as we have trained it for, hence lies the beauty of Machine Learning – Deep Learning techniques, the network does not understand or care whether the trainset was poetry or prose or code-bases. It works just as well.

FUTURE WORK

For future work purposes, we may want to expend more compute power and train the models for longer periods off time and see our predictions were accurate or not. We may want to apply these models into other applications where it might prove useful.

SUMMARY

In conclusion we want to put forth the ideas we discussed in this report. We talked briefly about Artificial Intelligence, Natural Language processing, and how they tie together. Machine Learning- Deep Learning architectures – RNN or recurrent neural networks and their sub-forms such as LSTM or Long-short-term-memory neural networks. Where these LSTM's are useful and why using these algorithms in nlp and to induce text-generative models are effective tools. We performed experiments on a corpus – a work of prose by Tagore called "The hungry stone"- and analysed the results and made observations on outputs generated by our models.

And we learned, why, where and how to use these models more effectively to solve real-life research, industrial problems through implementation more LSTM networks. Finally, we would like to express our gratitude to Dr Charles Ling for giving us the opportunity to learn from and explore these ideas and concepts through this project work.

REFERENCES:

1.[1] - http://jmc.stanford.edu/articles/coglunch.html

2.[2]- http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

3.[3] -Figure-3 https://medium.com/@mitra.rudradeb/using-lstm-network-to-model-sales-process-part-ii-ca797eb45d54

4.[4] https://en.wikipedia.org/wiki/Long_short-term_memory

5.https://en.wikipedia.org/wiki/Recurrent_neural_network