Text Generation Using LSTM

[1]R. Divya, [2]M. Ruchitha Gowd, [3]P. Sindhura, [4]P. Amrutha

[1,2,3,4] UG Student, Department of Computer Science Engineering,

Dr K V Subba Reddy College Of Engineering For Women, Kurnool, Andhra Pradesh, India

Article Info

Page Number: 476-480

Publication Issue:

Vol. 71 No. 2 (2022)

Abstract

Due to their capacity to learn dependencies over time, long short-term memory (LSTM) units on sequence-based models are utilized in classification, question-answering systems, and translation tasks. By learning language models with grammatically stable syntaxes, LSTM networks are providing impressive results in natural language generation for text generation models. The network, on the other hand, does not learn about the context. Regardless of pragmatics, the network only learns the input-output function and produces text from a set of input words. There is no semantic consistency among the sentences that are generated because the model is trained in a context that does not exist. A context vector and a predetermined set of input words are used to train the proposed model to produce text. A context vector, like a paragraph vector, understands the sentence's semantic meaning (context). In this work, several approaches to extracting the context vectors are suggested. In addition to the input-output sequences, context vectors are also trained alongside the inputs when a language model is being trained. The model learns the relationship between the target word, the context vector, and the input words because of this structure. A well-trained model will generate text based on the provided context given a set of context terms. Two variants of the model—word importance and word clustering—have been tested in light of the nature of computing context vectors. The appropriate embeddings between various domains are also examined in the word clustering method. The results are judged by how closely the generated text matches the given context in terms of semantics

Article History

Article Received: *25 December 2021*

Revised: *20 January 2022*

Accepted: *24 February 2022*

Publication: *28 March 2022*

1. Introduction

Deep learning is a subfield of machine learning that is entirely based on artificial neural networks. Since neural networks are intended to imitate the human brain, deep learning is also a form of brain imitating. We don't have to explicitly program everything in deep learning.

Deep learning is not a novel concept. Since a few years ago, it has been around. It's getting a lot of attention right now because we didn't have as much processing power or data in the past. Deep learning and machine learning emerged as a result of the exponential increase in processing power over the past two decades. Neurons are the formal definition of deep learning. A single neuron in the human brain is made up of approximately 100 billion neurons, and each neuron is connected to thousands of its neighbors.

This begs the question of how these neurons can be recreated in a computer. As a result, we construct an artificial structure with nodes or neurons that we refer to as an artificial neural

net. In the hidden layer, there may be a lot of interconnected neurons representing both input and output values. Artificial intelligence (AI) is the simulation of human intelligence in machines that are programmed to think and act like humans. The term can also be used to describe any machine that can learn and solve problems like a human brain can.

The capacity of artificial intelligence to rationalize and take actions that have the best chance of achieving a particular objective is its best quality. Machine learning is a subset of artificial intelligence that is based on the idea that computer programs can learn from and adapt to new data on their own without the help of humans. This automatic learning is made possible by deep learning techniques, which take in huge amounts of unstructured data like text, images, or videos. Computer programs that use deep learning go through a process similar to how a child learns to identify a dog. In order to generate a statistical model as an output, each algorithm in the hierarchy applies a nonlinear transformation to its input. The process continues until the output is accurate enough to be considered acceptable. The name "deep" came from the number of processing layers that must pass through data.

The learning process in traditional machine learning is supervised, and the programmer must be very specific when telling the computer what it should look for to determine whether an image contains a dog or not. This is a relentless interaction called include extraction , and the PC's prosperity rate relies completely on the software engineer's capacity to precisely characterize a list of capabilities for canine. The program builds the feature set by itself without supervision, which is a benefit of deep learning. Not only is unsupervised learning quicker, but it is typically more accurate

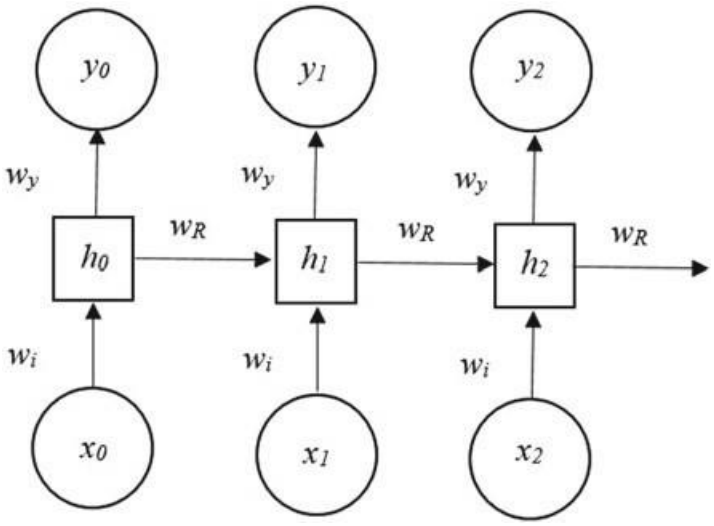

Fig.1 RNN Operation

2. Literature Review

One of the most widely used, non-generic models in the field of deep learning is the recurrent neural network (RNN). Character-level modeling is not a task that can be accomplished with the standard architecture of RNNs. As a result, a reworked version of RNN was proposed as a solution to the issues, and it used gated or multiplicative connections and the Hessian-free optimization technique to solve their training issues. A special type of recurrent neural

network known as a long short-term memory network (LSTM) was developed for the purpose of producing compound sequences. Predicting a single data point at a time is how these networks are built over a long period of time. The model that was demonstrated can effectively synthesize cursive handwriting in a wide range of styles. Neural checklist models are another way to generate text. By vigorously adjusting the interpolation into a model, this model generates output. The long short-term memory neural network architecture was utilized to examine RNN on a task of modeling an English and a French language in order to address these issues. RNN is difficult to train and it is doubtful to demonstrate the full potential of a given RNN model. It outperformed other presently available RNNs by approximately 8%. proposed two novel methods for the generation of text that first encode the contexts into a continuous semantic illustration and then use RNNs to decode the semantic illustration into text categorizations. The following research will demonstrate how using LSTMs can overcome the shortcomings of RNNs, such as vanishing gradients, which RNNs have their own problems with it.

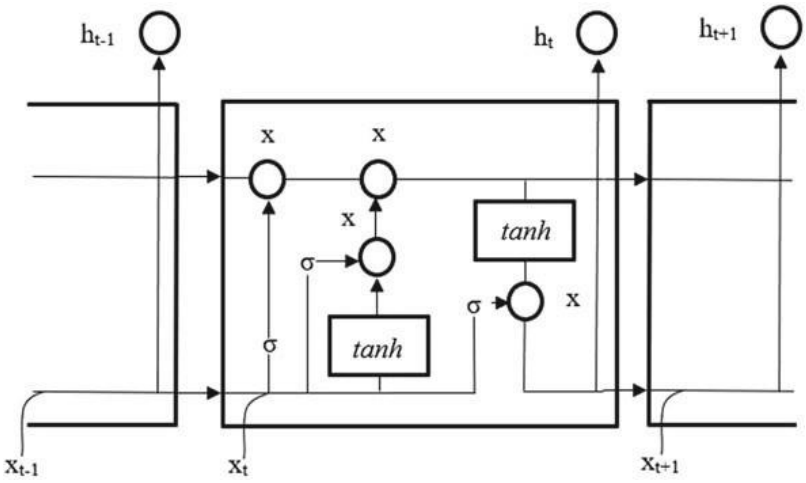

Fig.2 Working of a long short-term memory network

3. Proposed System

Python is a high-level scripting language that can be used for a wide range of tasks related to internet-related, system administration, and text processing. Its core language, unlike many other similar languages, is small and simple to learn, and modules can be added to perform virtually any task. Python is a true object-oriented programming language that runs on many different platforms. Python's standing as a great option for internet-based issues is further strengthened by the existence of a Java-only python interpreter.

With topics ranging from the fundamentals to the advanced (such as Web-scraping, Django, Deep-Learning, and others), this specialized Python tutorial will assist you in learning the Python programming language in the most effective manner. with instances. The following are some Python programming language facts:

At the moment, Python is the multipurpose, high-level programming language that is used the most. Both procedural and object-oriented programming styles are supported by Python. In general, Python programs are smaller than those written in other programming languages like

Java. Programmers have to type less, and the language's indentation requirement makes them always readable.

The Python programming language is utilized by nearly all tech giants, including Google, Amazon, Facebook, Instagram, Dropbox, Uber, and others. The desktop graphical user interface (GUI) Anaconda Navigator is a part of the Anaconda distribution. It lets users manage conda packages, environments, and channels without having to use command-line commands. Navigator is able to find packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages, and update them all at the same time. It is compatible with Linux, macOS, and Windows.

For data science and machine learning-related applications, Anaconda Navigator is a free and open-source distribution of the Python and R programming languages. Conda is a cross-platform, open-source package management system that can be installed on Windows, Linux, and macOS. Tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, and Visual Studio Code are included in Anaconda. We will use the Jupiter notebook and spyde for this project

```
</style>
<head>
<title>-- Music Genre Prediction for Spotify -- </title>
</head>
<body>
<div class='idiv'>

<br/>
<h1>Music Genre Prediction</h1>
<hr/>
<br/>
<form action="/predict" method="POST">

 <input class="form-input" type="text" name="Energy" placeholder="Enter the energy in music"><br>
 <input class="form-input" type="text" name="Loudness" placeholder="Enter the Loudness in music"><br>
 <input class="form-input" type="text" name="Speechiness" placeholder="Enter the speechiness in music"><br>
 <input class="form-input" type="text" name="Acousticness" placeholder="Enter the acousticness in music"><br>
 <input class="form-input" type="text" name="Instrumentalness" placeholder="Enter the instrumentalness in music"><br>
 <input class="form-input" type="text" name="Liveness" placeholder="Enter the liveness in music"><br>
 <input class="form-input" type="text" name="Valence" placeholder="Enter the valence in music"><br>
 <input class="form-input" type="text" name="Tempo" placeholder="Enter the tempo in music"><br>
 <input class="form-input" type="text" name="Duration" placeholder="Enter the duration of music in millisec"><br>
<button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
</form>
<br/>
<br/>
<br/>
```

Fig.3 Code Of The Project

Training and Test Data in Python Deep Learning As work with datasets, a deep learning algorithms works in two stages we usually split the data around 20%- 80% between testing and training stages. The achieved testing accuracy of 71.22% of the constructed model may be increased by increasing the no. of epochs and adding a greater quantity of layers/nodes to the current network.

```
Text Generated Using LSTM MOdel:

Seed:
" ere was not a moment to be lost: away went alice like the wind, and
was just in time to hear it say, "

Done.
```

Fig.4 Output Of The Project

4. Conclusion

Adding more data to be trained on Fine-tuning the model architecture, i.e., number of units, layers, etc., can improve the trained model's performance. Adjusting the parameters, such as the learning rate, activation function, epochs, and units, In addition, it has been demonstrated that LSTM networks are the most effective type of model currently available for performing prediction and classification on text-based data. The problem of vanishing gradient that is faced by conventional recurrent neural networks is successfully solved by LSTM. LSTM is a good model, but it is expensive to compute and needs a lot of processing power to fit and train, which means using GPUs. They are currently utilized in a variety of applications, including automated chatbots, voice assistants, sentiment analysis, and smart virtual keyboards. By applying transfer learning to the same problem domain and adding more layers and nodes to the network, future research may be able to surpass current LSTM models' model accuracy.

References

1. Reiter E (2007) An architecture for data-to-text systems. In: Proceedings of the eleventh European workshop on natural language generation. Association for Computational Linguistics
2. Saraswat S, Srivastava G, Shukla S (2018) Classification of ECG signals using cross-recurrence quantification analysis and probabilistic neural network classifier or ventricular tachycardia patients. Int J Biomed Eng Technol 26(2):141–156
3. Mikolov T et al (2010) Recurrent neural network based language model. In :Eleventh annual conference of the international speech communication association
4. Sak H, Senior A, Beaufays F (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling.
5. In: Fifteenth annual conference of the international speech communication association
6. Sutskever I, Martens J, Hinton GE (2011) Generating text with recurrent neural networks.
7. In: Proceedings of the 28th international conference on machine learning (ICML-11)
8. Graves A (2013) Generating sequences with recurrent neural networks. arXiv preprint arXiv: 1308.08502013