Applications of Machine Learning in Natural Language Processing and Time Series Forecasting

Siddharth Agrawal, Shaan Sheth, Shrey Sheth, Aarya Shah

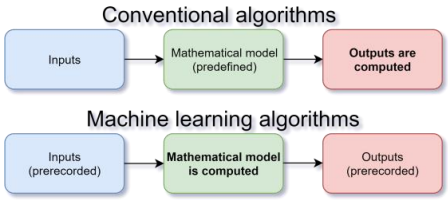
Abstract

Over the years, machine learning methods have developed to generate state-of-the-art models in several fields, with Natural Language Processing and Time Series Forecasting being some of the most prominent applications. An account of the history, structure, mathematics, advantages, and limitations of some machine learning techniques such as Markov chains, word2vec, Recurrent Neural Networks, Long Short-Term Memory, seq2seq, autoencoders, and transformers are discussed in this paper, with reference to their applications in Time Series Forecasting of stock prices and website-traffic, and various Natural Language Processing tasks such as autoregression, summarisation, downstreaming, classification, topic modelling, plagiarism detection, abstract generation, keywords extraction, etc.

Index Terms—Machine Learning, Natural Language Processing, Time Series Forecasting, Neural Networks, Deep Learning

Introduction

Artificial neural networks are a type of machine learning algorithm. Machine learning is a branch of discreet mathematics and computer science. Typical algorithms give an output based on predefined mathematical model and programming, and any input(s). Machine learning algorithms[1] take pre-recorded inputs and outputs to generate the mathematical model which can then be used to estimate outputs based on other inputs.



Defining the steps or formulating an algorithm to take inputs as previous stock data and information to predict future stock prices is extremely difficult as they have a very complex relationship. Neural networks are extremely versatile at determining these complex relationships between inputs to outputs.

In this investigation, we will be exploring several Machine Learning models such as SVM, and Markov Chain Model.

We will also be exploring several types of Neural Networks: Deep Artificial neural networks, recurrent neural networks, long-short term memory neural networks, Temporal Convolution Networks, and Transformers. Knowledge and understanding of each of the networks serves as a basis to understand the next network.

Mathematics of such machine learning models will be explored and then applied for the following tasks:

Using machine learning models and applying them on datasets such as: IMDB, Yelp-2, Yelp-5, Amazon-2, Amazon-5, 20newsgroups

Using machine learning models and applying them on several time series prediction tasks such as: electricity power consumption, web-traffic forecasting, stock-price prediction

Background

[1]Machine learning algorithms: Application of artificial intelligence in providing systems the ability to automatically generate, learn, and improve from experience by themselves, without explicit programming.

Time Series Forecasting includes various applications where the future data needs to be predicted. This includes climate and weather forecasting models, stock price prediction models, web-traffic prediction models, machine failure prediction models, etc. Time Series Forecasting is very varied and different models are useful in different applications. However, many of the models used overlap with models used in NLP therefore allowing us to explore more applications without the need of coding additional models.

There are many notations used within the report that are used in computer science, especially, artificial neural networks' nomenclature which may seem to be plagiarised but are, in fact, the works of the author.

Natural Language Processing is the application of computational techniques for natural language, speech, and text. It is used for processing, analysing, classifying, summarising, topic modelling, plagiarism detection, plagiarism rewriting, grammar checking, spell checking, etc. of text.

Most important aspects of an algorithm are its efficacy and efficiency. The predictive capabilities of the model (efficacy) have to be maximized, but the amount of time it takes to optimize the model (efficiency) has to be minimized. In order to increase efficiency, the no. of calculations, and the complexity of the calculations, calculated by the computer has to be reduced.

Relevant Machine Learning models include SVM, Markov Chain Model, ANN, RNN, LSTM, TCN, Transformers.

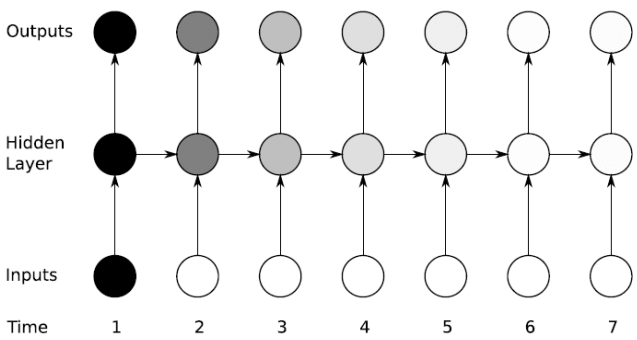
Motivation

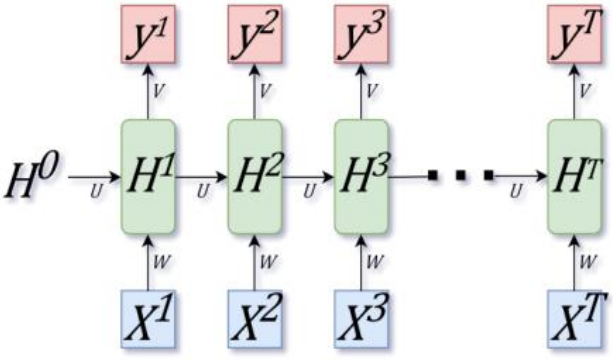
While plenty of literature and pretrained models as well as architectures are available in abundance. The in-depth mathematical and computational understanding of models used in NLP and TSF is lacking. This paper will be demystifying these models by delving deep into the mathematics and computations behind the ML models used for such applications. The paper will cover the basic ML implementations such as Markov Chains and networks such as Feed Forward Networks and then move on to more complex models such as Recurrent Neural Networks, Long Short-term memory Neural Networks, Seq2Seq networks, and Transformers.

Literature Survey

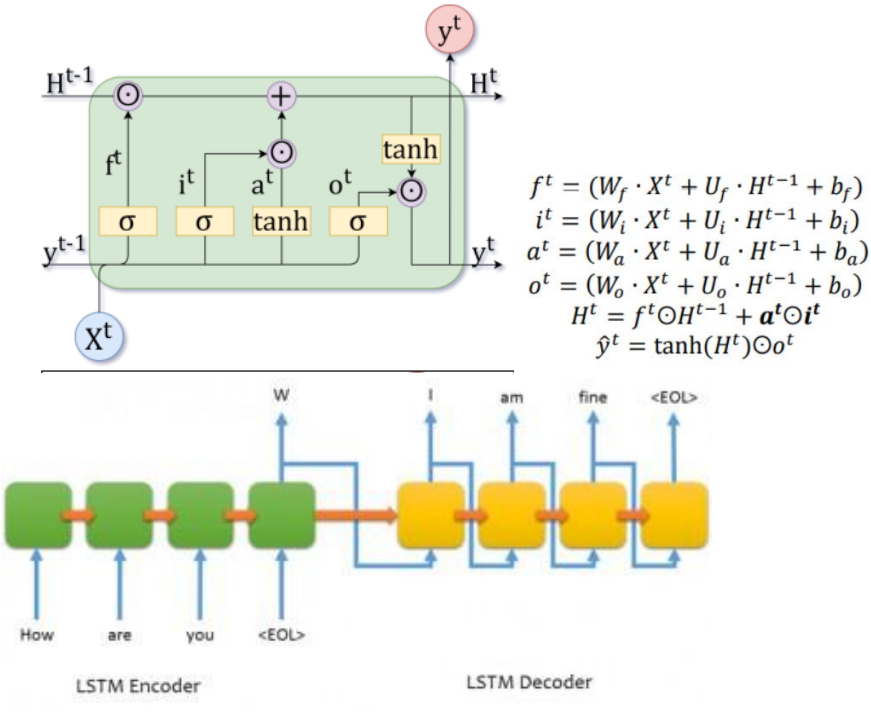
Conventional strategies of stock prediction such as news, buy and hold, martingales, momentum, mean reversion, etc. predict stocks with limited efficacy. With improvements in technology, stock trading has been transferred to automated computer predictions using ML.

RNN are the most basic form of sequence based neural network but it suffers from vanishing gradients.





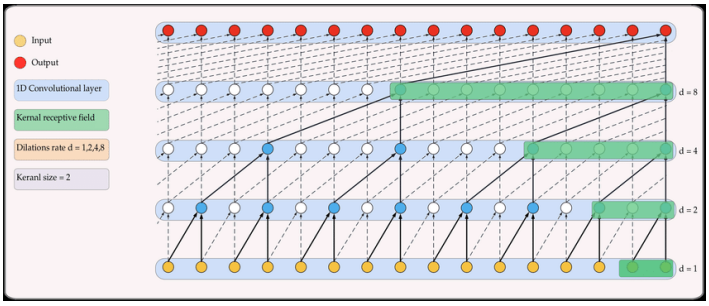
To address vanishing gradients problem, many NLP models used LSTM networks. Such networks had longer memory than RNN networks. Furthermore, they could be used on large documents and text sequences due to their recursive nature as the memory requirement did not depend on input sequence length.



$$f^t = (W_f \cdot X^t + U_f \cdot H^{t-1} + b_f)$$
$$i^t = (W_i \cdot X^t + U_i \cdot H^{t-1} + b_i)$$
$$a^t = (W_a \cdot X^t + U_a \cdot H^{t-1} + b_a)$$
$$o^t = (W_o \cdot X^t + U_o \cdot H^{t-1} + b_o)$$
$$H^t = f^t \odot H^{t-1} + \boldsymbol{a^t} \odot \boldsymbol{i^t}$$
$$\hat{y}^t = \tanh(H^t) \odot o^t$$

Later seq2seq models developed by google which produced higher accuracy and efficacy in various NLP and time series tasks. It used an encoder and a decoder model. One of the LSTMs would encode the text into another language and the other would try to decode it. Though it was first designed for translation and text2speech or speech2text tasks, it outperformed conventional LSTMs in other NLP tasks as well. This was because this architecture facilitated unsupervised learning.

New architectures that used convolution on time series were also effective. Called Temporal Convolutional Networks (TCN), they also outperformed LSTM in tasks that required very long sequences. The text or input sequence is divided into smaller sequences, which are then pooled together to get a new set of sequences that accounted for larger percentage of the overall input sequence. This process is repeated till the input sequence is sufficiently small and then mapped to outputs after through a dense layer.



They addressed vanishing gradient problem that still exists in LSTM (though to a much lesser degree) by using an approach similar to divide-and-conquer.

They exhibit longer memory than recurrent architectures like RNN, LSTM, GRU and are able to perform better on vast range of tasks such as Word-level PTB, MNIST, Adding Problem, Copy Memory, etc.

They also allowed for parallelism which recurrent architectures do not (as they need the output from the first input/timestep in order to feed the data into itself recursively and produce outputs for future timesteps).

2017 was a great year for NLP progress. With papers such as “Efficient Estimation of Word Representations in Vector Space” and “Attention Is All You Need” being published. “Efficient Estimation of Word Representations in Vector Space” covered a new architecture called Transformers that could be used to encode vector representations of words. This technique, commonly known as word2vec in NLP nomenclature became very popular and effective as it could represent large vocabulary as a vector space which reduced the dimensionality and memory issues with one-hot encoding. “Attention Is All You Need” served as a landmark paper as it changed our conventional understanding of NLP models away from TCN and LSTM networks onto Transformer and self-attention based networks.

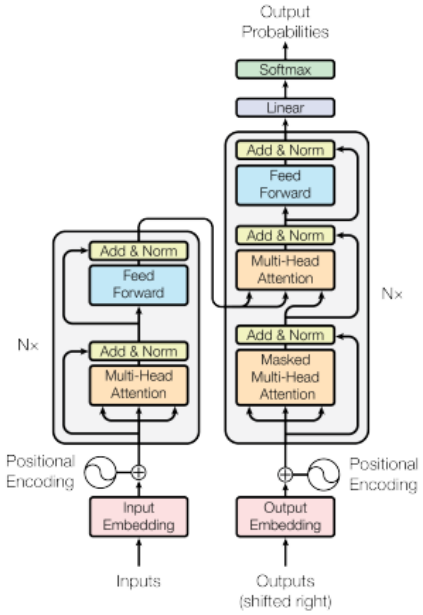


Figure 1: The Transformer - model architecture.

Transformer based self-attention networks have several advantages over older network architectures.

- Reduced worst case complexity for path length thus reducing vanishing gradient problem.
- Increased parallelisation capabilities due to its non-recursive nature.
- Bi-directionality: words can be placed in their context as the model has access to the words both before and after the token/word to be predicted. Therefore, understanding textual context better and increasing accuracy in benchmarks.
- Is used in all current generation models including GPT2, XL-Nets, BERT, RoBERTa, GPT3, etc.
- Disadvantage: Cannot be used for long text sequences as the space and computational complexity is O(n^2). Though a potential remedy is sparse attention networks.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

References

"A Primer on how to optimize the Learning Rate of Deep Neural Networks", by Timo Böhm, *Towards Data Science*, https://towardsdatascience.com/learning-rate-a6e7b84f1658 Accessed on: 19 August, 2020.

"Activation Functions in Neural Networks", by Sagar Sharma, *Towards Data Science,* Sep 6, 2017. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 Accessed on: 4 September, 2020.

"Backpropogating an LSTM: A Numerical Example", by Aidan Gomez, *Medium*, Apr 19, 2016. https://medium.com/@aidangomez/let-s-do-this-f9b699de31d9 Accessed on:19 September, 2018.

"Definition of Outer Product", *Chegg Study*, 2018. https://www.chegg.com/homework-help/definitions/outer-product-33 Accessed on: 7July, 2020.

"Estimating an Optimal Learning Rate for a Deep Neural Network", by Pavel Surmenok, *Towards Data Science*, Nov 13, 2017 https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0 Accessed on: 17 August, 2018.

"How to Multiply Matrices," *Advanced Math is Fun*, 2017, https://www.mathsisfun.com/algebra/matrix-multiplying.html Accessed on: 4 August, 2020.

Indian Mutual Funds Handbook: A Guide for Industry Professionals and Intelligent Investors, by Sundar Sankaran, Vision Books Pvt. Ltd., 2003, Accessed on 17 September, 2018.

"Learning Rate Tuning and Optimizing", by Chaitanya Kulkarni, *Medium,* Feb 19, 2018, https://medium.com/@ck2886/learning-rate-tuning-and-optimizing-d03e042d0500 Accessed on: 23 August, 2020.

"Machine Learning week 1: Cost Function, Gradient Descent and Univariate Linear Regression", by Lachlan Miller, *Medium,* Jan 10, 2020 https://medium.com/@lachlanmiller_52885/machine-learning-week-1-cost-function-gradient-descent-and-univariate-linear-regression-8f5fe69815fd Accessed on: 11 July, 2020.

The Matrix Cookbook by Kaare Brandt Petersen, Michael Syskind Pedersen, January 5, 2005https://www.ics.uci.edu/~welling/teaching/KernelsICS273B/MatrixCookBook.pdf Accesse d on 9 October, 2020

"Neural Networks Demystified [Part 2: Forward Propagation]", by Welch Labs, *Youtube,* Nov 7, 2014. https://www.youtube.com/watch?v=UJwK6jAStmg Accessed on: 30 July, 2020.

"Neural Networks Demystified [Part 3: Gradient Descent]" by Welch Labs, *Youtube,* Nov 21, 2014. https://www.youtube.com/watch?v=5u0jaA3qAGk Accessed on: 30 July, 2020.

"Only Numpy: Deriving Forward feed and Back Propagation in Long Short Term Memory (LSTM) part 1", by Jae Duk Seo, *Towards Data Science*, Jan 12, 2020.

https://towardsdatascience.com/only-numpy-deriving-forward-feed-and-back-propagation-in-long-short-term-memory-lstm-part-1-4ee82c14a652 Accessed on: 15 July, 2020.

"The Artificial Neural Networks handbook: Part 1", produced by Jayesh Bapu Ahire, *Data Science Central*, August 24, 2018 . https://www.datasciencecentral.com/profiles/blogs/the-artificial-neural-networks-handbook-part-1 Accessed on: 5 September, 2020.

"Understanding LSTM Networks", by Felix Gers, et.al., *Colah's Blog,* August 27, 2015. http://colah.github.io/posts/2015-08-Understanding-LSTMs/ Accessed on: 5 September, 2018. 21 July, 2020.

"What is the sigmoid function, and what is its use in machine learning's neural networks? How about the sigmoid derivative function?" by Vinay Kumar R *Quora,* Aug 24, 2017. https://www.quora.com/What-is-the-sigmoid-function-and-what-is-its-use-in-machine-learnings-neural-networks-How-about-the-sigmoid-derivative-function Accessed on: 7 September, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, Aidan N, Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. ArXiv.org. https://arxiv.org/abs/1706.03762

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. ArXiv.org. https://arxiv.org/abs/1301.3781

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv.org. https://arxiv.org/abs/1810.04805

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. ArXiv.org. https://arxiv.org/abs/1906.08237

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv.org. https://arxiv.org/abs/1907.11692

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. ArXiv.org. https://arxiv.org/abs/1901.02860

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. ArXiv.org. https://arxiv.org/abs/2004.05150

Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). DocBERT: BERT for Document Classification. ArXiv.org. https://arxiv.org/abs/1904.08398