International Journal of Computing Sciences Research (ISSN print: 2546-0552; ISSN online: 2546-115X)
Vol. 7, pp. 1076-1091
doi: 10.25147/ijcsr.2017.001.1.91
https://stepacademic.net

Short Paper*

An Enhancement of Long-Short Term Memory for the Implementation of Virtual Assistant for Pamantasan ng Lungsod ng Maynila Students

Draillim Xaviery F. Valonzo
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
dxfvalonzo2018@plm.edu.ph
(corresponding author)

Jose Ramon M. Jasa
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
jrmjasa2018@plm.edu.ph

Mark Christopher R. Blanco
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
mcrblanco@plm.edu.ph

Khatalyn E. Mata
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
kemata@plm.edu.ph

Dan Michael A. Cortez
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
dmacortez@plm.edu.ph

Date received: November 15, 2021
Date received in revised form: January 21, 2022; January 23, 2022
Date accepted: January 23, 2022

Recommended citation:

Valonzo, D. X. F., Jasa, J. R. M., Blanco, M. C. R., Mata, K. E., & Cortez, D. M. A. (2023). An enhancement of long-short term memory for the implementation of virtual assistant for Pamantasan ng Lungsod ng Maynila students. *International Journal of Computing Sciences Research*, *7*, 1076-1091. https://doi.org/10.25147/ijcsr.2017.001.1.91

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

**Special Issue on National Research Conference in Computer Engineering and Technology 2021. Guest Editor: Dr. Luisito Lolong Lacatan, PCpE (Laguna University, Philippines). Associate Editors: Dr. Nelson C. Rodelas, PCpE (University of the East-Caloocan City, Philippines), and Engr. Ana Antoniette C. Illahi, PCpE (De La Salle University, Manila, Philippines).*

Abstract

Purpose – Natural Language Processing is an aspect of Artificial Intelligence that focuses on how technology can understand words, derive meaning from them, and return a meaningful and correct output. Therefore, it is used in the making of Virtual Assistants today. Training virtual assistants require long temporal dependencies and sequence-to-sequence classification. This study will be used to create a possible algorithm that will enhance the performance of a possible virtual assistant designed for PLM students and faculty members.

Method – LSTM will be used to train the model to address these concerns. However. The LSTM algorithm faces the problem of slow computing speed and high computation costs. To address this the researchers implemented TensorFlow XLA to the model to optimize the computation costs in the problem.

Results – Though the number of matrices exploded from 934 to 30000, the training can show slight improvement both in memory, CPU (Central Processing Unit) utilization, and time reduction. At 50 epochs, training the model with XLA has shown a time decrease of 8 minutes and can save at most 500 megabytes of memory.

Conclusion – XLA has proven that it has helped the LSTM algorithm in terms of its usage in memory, utilization of CPU, and overall speed of training, especially in longer processes.

Recommendations – The researchers recommend using XLA in the context of pruning and the effect of pruning paired with XLA to maximize the performance of the model.

Practical Implication – This would allow a much more efficient and cost-friendly training of the model when feeding it new data to be used for virtual assistant designed for PLM students and faculty members.

Keywords – LSTM, NLP, RNN, Virtual Assistants, Artificial Intelligence, Neural Networks

INTRODUCTION

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that is concerned with methods that allow computers to understand human speech and respond

to it, accordingly, combining computational linguistics and statistics (What Is Natural Language Processing?, 2020) This aims to read a human speech and words and decode it in a valuable manner (*A beginner's introduction to Natural Language Processing*, 2021), allowing us to automate processes that concern communication and human words. This is the reason NLP became one of the most popular branches of AI as it was used through advertisements and Google's search engines (Mills, 2018).

One of the technologies on the rise due to NLP is the Intelligent Virtual Assistants (IVA), which are developed entities that interface with people in a human way (*What Does Intelligent Virtual Assistant Mean*, 2021), mostly providing customer service solutions. Virtual Assistants can enhance meeting experiences and companies' productivity ensuring that virtual assistants had eliminated excess time in scheduling, reduced translation cost and improved team efficiency by storing notes and transcripts, raising the need for more functional IVA such as Cortana, Siri, Nina, and Google Assistant (Lazar, 2021). This led to the rise of NLP algorithms needed to produce such software. In 2018, Google developed Bidirectional Encoder Representations from Transformers (BERT) which analyses languages both from right to left and left to right and learns in small datasets with great accuracy (McGregor, 2020), performing well to the text-to-speech problems. These techniques are coupled with the use of a neural network for pattern recognition (Klie, 2014). Inputs are then processed and compared to the patterns present in the neural network and then their probabilities are weighed. The highest probability pattern in the neural network will be the answer passed as the most probable response to the query of the user.

Given that Neural Networks are used in the creation of the Virtual Assistant, the researchers used the Long-Short Term Memory neural network, which is a type of RNN, as an effective solution when it comes to sequential data. The LSTM architecture is run by the idea of a memory cell that keeps its state constant despite the time, and nonlinear gating units, which control the flow of information through the entrances and exits of the network (Greff et al., 2017). One of the most important parts of the LSTM network, which is one of the key features on why it is an improvement of the standard RNN, is the forget gate. A forget gate is a mechanism of the LSTM network which oversees forgetting information. Van der Westhuizen and Lazenby (2018) stated that forgetting information, prevents the network from growing indefinitely and the network avoids the possibility of being broken down. LSTM has been used for the advancement of several aspects of technology, such as handwriting recognition, speech synthesis, and audio and video analysis, among others (Greff et al., 2017).

However, the LSTM neural network also comes with its downsides. Despite being a network that is deemed as an improvement from the standard RNN because of its added features for solving long-term memory problems, it also comes at the cost of the model size being of a bigger scale. This is where the first problem arises. LSTM needs a big memory and a large bandwidth for its processes to be executed properly, which can make the process slow and time-consuming (Wang et al., 2019).

This study may serve as an initiative to further enhance the development of the algorithm which will be used for the possible creation of the Virtual Assistant to be used by PLM's students, faculties, stakeholders, and other concerned parties.

LITERATURE REVIEW

Natural Language Processing and Virtual Assistants

Natural Language Processing is defined as a list of computational techniques used to analyze and represent naturally occurring texts at one or more linguistic analysis levels to make human-like interactions that can be used in various aspects. NLP was born out of the contributions of Linguistics (form and structure of language and the discovery of language universals), Computer Science (building internal data representations and efficient structure processing), and Cognitive Psychology (using language usage as a medium to see how the human cognitive process works, and how language can be used in a psychologically plausible manner) (Liddy, 2001).

Natural Language Processing has two distinct focuses, which are language processing and language generation. Language processing is the analysis of language to accomplish the goal of achieving meaningful representation, while Language Generation is the production of language based on what they can pick up from that representation. Another distinction can be said with the understanding of language and the understanding of speech, wherein Oral Language is the beginning of Speech understanding and the end of Speech generation. Hence, these two fields require the help of phonology and acoustics. The understanding of speech will focus on how the system will pick up a certain sound of language in the form of acoustic waves which are then transcribed into words that are familiar to people. When this happens, the processing which is applied to written text can also be applied here (Liddy, 2001).

Natural Language Processing can be explained by using the approach of the "levels of language," which states that human language processing can be made into a sequence wherein each step is followed by another step in a sequential manner. Psycholinguistic research suggests that this is dynamic because the different levels can interact differently, while introspection states that We regularly use information from what is traditionally thought of as a higher level of processing to aid with lower-level analysis (e.g. When a person reads a book about computers, the pragmatic knowledge that book he/she is reading is about computers will be used when a particular context that has several meanings or contexts is seen, and the word is interpreted in the context of computers. The main point of the levels of language is that every level of language will convey meaning and that since humans use all levels of language to gain understanding, the more levels of language a system will utilize, the more capable its NLP system is (Liddy, 2001). These levels are, namely, Phonology (interpretation of speech sounds in words), morphology (componential nature of words), Lexical (Meanings of Individual Words), Semantics (The Meanings of a sentence based on the interactions of the words that make the sentence),

Discourse (interpretation of a sentence based on the connections it has on component sentences), and Pragmatic (the use of the language in a certain situation or context) (Liddy, 2001)

Natural Language Processing has been used in a variety of aspects and principles to help people integrate text and speech into technology. It is the foundation of AI-based chatbots because of their ability to complete tasks that humans do. Because of NLP, chatbots now can assess an input, retrieve its context, and submit a meaningful and correct response (Ayanouz et al., 2020). It has been known to be used in the context of formal education (Molnar & Szuts, 2018), health care and medicine (Ayanouz et al., 2020), Online Business and Marketing (Husak et al., n.d.). It has also been applied in the aspects of Information Retrieval, Information Extraction, Question Answering, Machine Translation, and Dialogue Systems (Liddy, 2001).

The Long-Short Term Memory Algorithm

Compared to data mining models, text features work by sequentially relating to each other. In cases like this, it is essential to use sequence-to-sequence classifiers to solve this problem (Sri, 2021). Because of this reason, the researchers chose the Long-Short Term Memory algorithm in training the model.

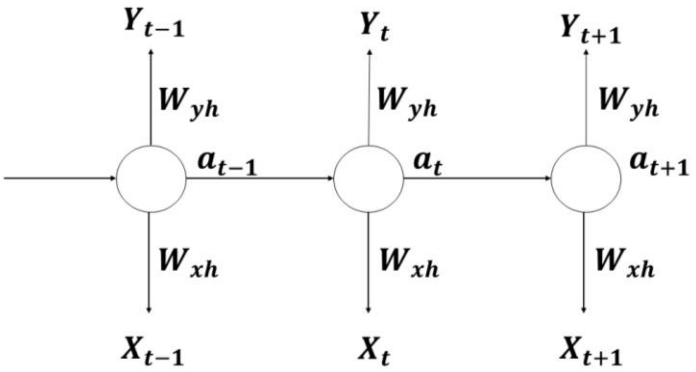

Figure 1. A Basic Recurrent Neural Network Architecture

As illustrated in Figure 1, the hidden layer xt-1 is outputted to the next layer along with the next time input X_t. The third layer then takes the hidden layer input present in X_t and X. This is where the problem of the RNN is now apparent. This type of network has no way to know what data or inputs to "remember" or which data or inputs to "forget." In sentences, some words are not used to affect the probability of other words occurring. (e.g., in the sentence "I eat Potato Chips every day," the dependence of the word "every day" is higher with the word "eat" than with the words "I" or "Potato Chips." It is also apparent that The Recurrent Neural Networks suffer from the Vanishing Gradient Problem, wherein as time passes and the gradients are moved from the upper to the lower layers, the gradients start to get smaller and "vanish," therefore resulting in its non-contribution to the existing algorithm (Sri, 2021).

This is where the LSTM algorithm shines. LSTM is a variant of the RNN, wherein it is an algorithm designed to learn data representations through sequencing (Mirwan et al., 2018), such as in the case of NLP in the aspect of generating text (Sri, 2021). The LSTM algorithm and its variants are used to capture long-term dependencies, which in turn is used to achieve high accuracy, hence, it is used in many applications (Nan et al., 2020).

The common structure of an LSTM Layer is shown in figure 2:

$$\boldsymbol{i_t = \sigma(W^i X_t + U^i h_{t-1} + b^i)} \quad (1)$$
$$\boldsymbol{f_t = \sigma(W^f X_t + U^f h_{t-1} + b^f)} \quad (2)$$
$$\boldsymbol{o_t = \sigma(W^o X_t + U^o h_{t-1} + b^o)} \quad (3)$$
$$\boldsymbol{\check{c}_t = \tanh(W^o X_t + U^o h_{t-1} + b^o)} \quad (4)$$
$$\boldsymbol{c_t = f_t \otimes c_{t-1} + i_t \otimes \check{c}_t} \quad (5)$$
$$\boldsymbol{h_t = o_t \otimes \tanh(c_t)} \quad (6)$$

Figure 2. A Common LSTM Layer Structure

As shown in Figure 2, the equations from 1 to 4 are the basic gates present in a layer of the LSTM algorithm. These 4 layers are called the input gate, the forget gate, the output gate, and the candidate memory. Through these four gates, the LSTM algorithm can achieve the capturing of long-term temporal dependencies. W^j and U^j ($j = i, f, o, c$) are the matrices of the weights, c_t and h_t denote the time-related values which are used for the transferring of information every step of the way. $\otimes$ is denoted as the element-wise multiplication while σ and tanh are the nonlinear activation functions (Nan et al., 2020).

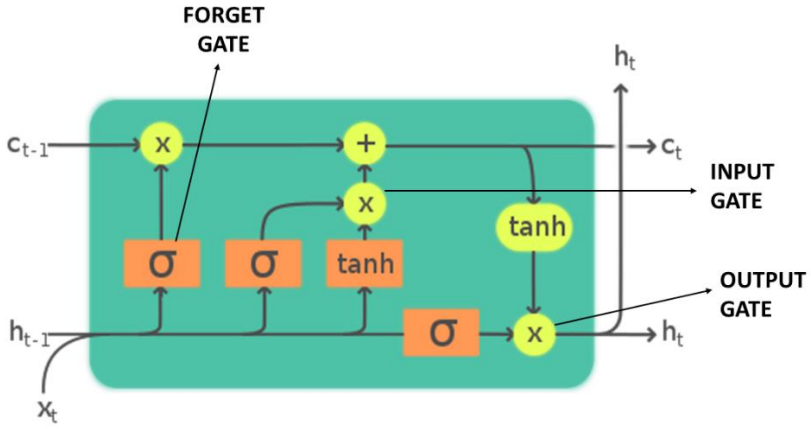

Figure 3. The Gates of the LSTM Algorithm

As shown in Figure 3 (Wikimedia Foundation, 2021), the LSTM algorithm is made up of cells that output two kinds of values: hidden states and cell states. The hidden state is the modified cell state, which is used as the output from the cell, while Cell states are the

memory manipulated by the three gates of the LSTM algorithm to identify what information to remember and forget (Sri, 2021).

The three gates are, namely, the Forget gate, the Input gate, and the Output gate. Figure 4 (Sri, 2021) Illustrates the Forget Gate which is the part where modifications are made wherein the cell value is modified to forget a certain part of the information in the input. This process is made possible through the combining of the previous hidden states with the current value and squishing these values through the sigmoid function. When this array of 0 to 1 number is used, it serves as a forget gate in the case where it is multiplied by the last cell's cell states (Sri, 2021).

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

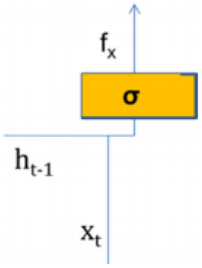

Figure 4. The Forget Gate

As shown in figure 6 (Sri, 2021), The Input gate determines what information must be added to the cell state. With sigmoid and tan activations, it is a weighted multiplication of the last hidden and current input. The first process is that the last cell's hidden state is multiplied by current values and then squashed by a sigmoid function. The last cell's hidden state will multiply with current cell values, but they will be squashed by a tan function. The two vectors are multiplied, then the cell value is added. Compared to the forget gate, wherein the multiplier values range from 0 to 1 in the previous cell state, input gates are additive, meaning that the outputs of the set of operations are added with the cell state (Sri, 2021).

$$i_t = \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right)$$
$$\tilde{c}_t = \sigma_h\left(W_c x_t + U_c h_{t-1} + b_c\right)$$

Figure 5. The Input Gate Equation

The final gate, as shown in Figure 5 (Sri, 2021) is a product of c_t and i_t. This is now added to the modified cell state

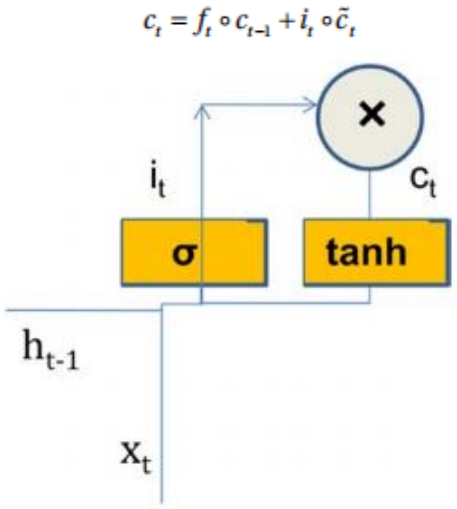

Figure 6. The Input Gate and The Input Gate Illustration

Lastly, Figure 7 (Sri, 2021) shows that the Output gate is the final gate that determines what will be considered as an "output" from the cell state. These values have been "filtered," meaning that not all the values are taken as output and are taken out of the cell. The output state is the result of multiplying the last cell's hidden state and the current output (Sri, 2021). This will become the output gate's "filter" when it becomes squished with the sigmoid function. This cell state now gets squished into a tan function. This cell state will then be modified by the "filter" made in the previous process (Sri, 2021).

$$o_t = \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right)$$
$$h_t = o_t \circ \sigma_h\left(c_t\right)$$

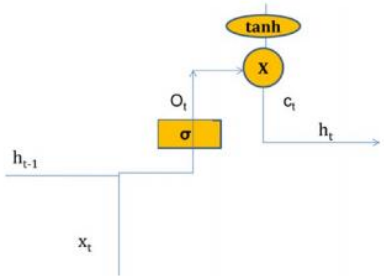

Figure 7. Output gate Equation and Output Gate Illustration

The Problem with LSTM

The Long-Short Term Memory Algorithm, though powerful, tends to encounter problems regarding space and time efficiency (Hou et al., 2019). The processes made in the input, the hidden cell, the output gate, and then looping it takes to churn out efficient outputs are heavy in processing space, especially in the case of devices that are lacking in resources and power to run these processes (Wang et al., 2018). The LSTM algorithm can learn from experiences to categorize, process, and predict time series when there are long time lags of unknown size between key events in a system. It consists of blocks that remember values for a random time frame. However, because of the LSTM's multiple

nodes filled with recurrent and cross-connections, it gets harder and harder to train as the layers pile up and the sequences are increased (Lyu & Zhu, n.d.).

The LSTM's computations involved in their model need enormous amounts of data which cannot all be stored on the static random-access memory (SRAM) because of the sheer size of the processes. The dynamic random-access memory (DRAM), a low-cost and high-capacity external memory are required in the hardware architecture for LSTM inference. Access to DRAM, however, take more than two orders of magnitude of energy. It also takes more time for the data to travel from one point to another in DRAM. Data exchanges between the SRAM and the DRAM will result in the high consumption of space, time, and power (Wang et. al.,2019). Therefore, the LSTM algorithm needs a lot of computational power and substantial energy and memory resources (Silfa et al., 2018).

METHODOLOGY

To address the high-compute characteristics of training the LSTM model in Keras the researchers used the Accelerated Linear Algebra (XLA) in training the model. A kind of optimization in TensorFlow models that compiles the TensorFlow models into optimized kernels that are fitted for a given architecture (Singh, 2020). Activating XLA will allow TensorFlow to do optimization in the LSTM model such as:

$$i_t = sigm(\theta_{xi} \times X_t + \theta_{hi} \times h_t - 1 + b_i) \quad \textit{Equation 1}$$

$$ft = sigm(\theta_{xf} \times X_t + \theta_{hf} \times h_t - 1 + b_f) \quad \textit{Equation 2}$$

$$o_t = sigm(\theta_{xo} \times X_t + \theta_{ho} \times h_t - 1 + b_O) \quad \textit{Equation 3}$$

$$g_t = sigm(\theta_v \times X_t + \theta_{hg} \times h_t - 1 + b_t) \quad \textit{Equation 4}$$

$$c_t = f_t \cdot c_t - 1 + i_t \cdot g_t \quad \textit{Equation 5}$$

$$h_t = o_t \cdot tanh(c_t - 1) \quad \textit{Equation 6}$$

The equations above are the implemented optimizations in XLA to maximize the optimization capacity of matrix multiplication in each LSTM cell (Chadha & Siddagangaiah, n.d.; Patil et al., 2020). As we can infer there are optimizations done in forget, input, and output gates as well as the hidden states in each cell done to crunch the multiplication matrix in LSTM as shown in equations 1, 2,3, 4, 5, and 6 (Chadha & Siddagangaiah, n.d.). Though XLA also means that the model will reshape the matrix that is inputted in the model, in our case a (13,) shape will turn into (1,), and the training batch will explode from 938 to 30000 due to the tiling method that XLA applies to the model. This is to handle the sequential order of the memory faster due to the reduction of a matrix that will be handled by the model as shown in Figure 8 (Tiled Layout | XLA |, 2021).

0,0	0,1	0,2	0,3	0,4	
1,0	1,1	1,2	1,3	1,4	
2,0	2,1	2,2	2,3	2,4	

0	2	4	6	8	10	12	14
1	3	5	7	9	11	13	13

Figure 8. Data Memory Tiling Reshape Illustration

Another optimization done by the XLA is the modification done on the bitwise operation of the softmax operation by fusing the point-wises subtraction from maximum and exponential operations to form a combined kernel allowing both subtraction and exponential operations without reading/writing from memory every time as shown in Figure 9 (Chadha & Siddagangaiah, n.d.), which reduces the process that is done by the machine when softmax is used as an activation function.

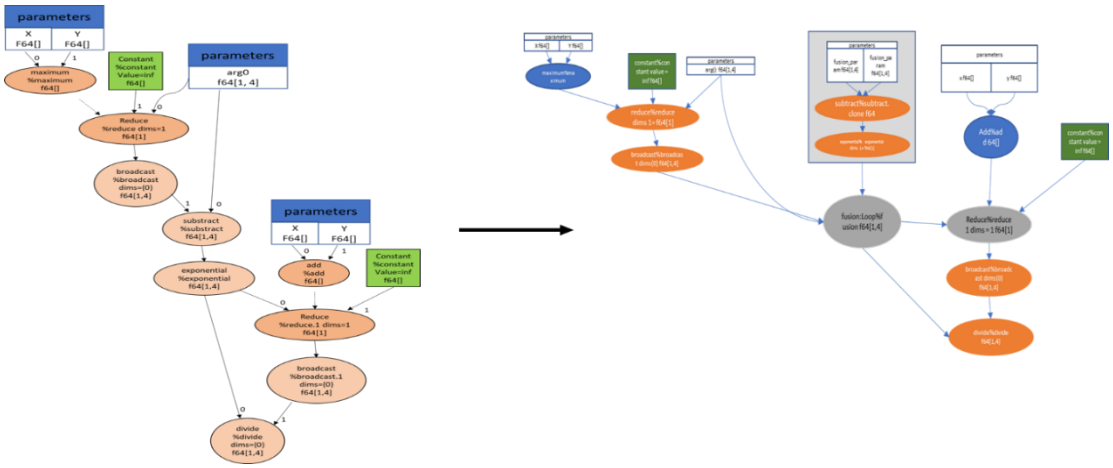

Figure 9. Softmax Before and After XLA illustration

Implementing these modifications, XLA can optimize some of the heavy operations inside the model.

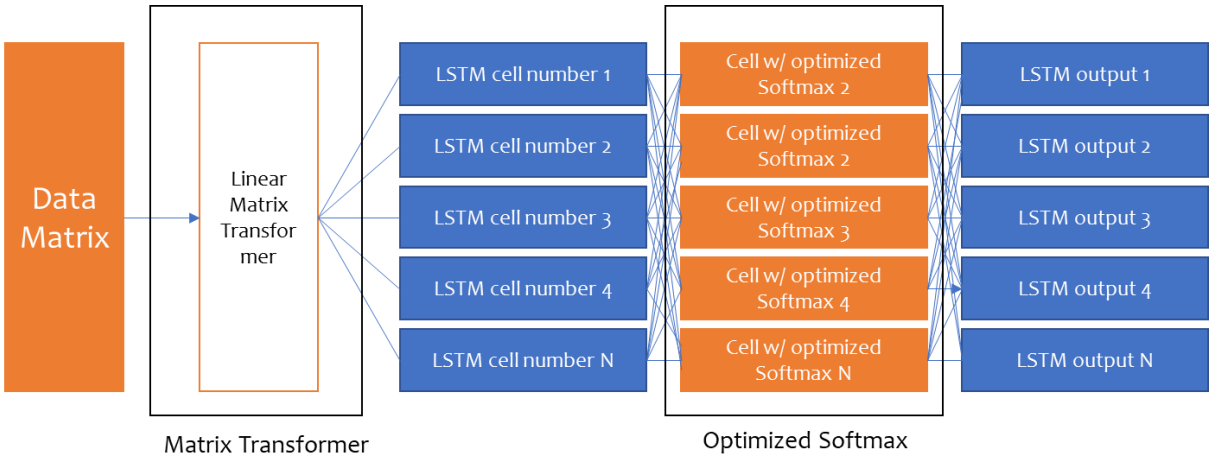

Figure 10. Overall LSTM architecture during XLA activation

As presented in Figure 10, which is the proposed model with the XLA optimization, the model will first transform the multi-layered matrix into a single layer matrix. The model will then use optimized softmax operation as the activation function to transform the data and give the probability of each data which in this case the texts trained in using the model. The model used the PLM plus Cornell Movie Conversation Model as the dataset to train the model which the Cornell Edu distributes. The dataset has in whole 220567 conversations however we only took 29,000 of those data with 1,000 from the PLM dataset which in all consists of 30,000 datasets. The dialogue in Cornell datasets has attributes like lineId, CharacterId, MovieID, character name (Patil et al., 2020), the test however the test only included the lineID and the text since the rest of the attributes is not significant to the test. The attributes were separated by the '++$++' but is it cleaned on the when used.

To gauge the speed of the training of the model, on the other hand, the researchers created a Callback class that will start counting the seconds of training in each epoch through self.starttime=time() and will append the total seconds it took to finish training through self.logs.append(time()-self.start time). The test will first start with 10 epochs both for the non-XLA and XLA model as a base test and then will increase it to 50 epochs as referred to (Patil et al., 2020) with a batch size of 1 to observe the model if accuracy is maximized.

RESULTS

Table 1 is the pre-test status of the machine before running the data between different states. The table below shows that there are slight differences between the states however they are not that significant since they will be deducted from the actual data in testing

Table 1. Pre-Test Machine Status

State	Memory (in Gb)	CPU	Disk memory
10 epochs w/o XLA	2.8	1%	0%
10 epochs w/ XLA	2.1	1%	0%
50 epochs w/o XLA	2.4	2%	20%
50 epochs w/ XLA	3.7	10%	5%

Table 2 shows the difference between the running status of the machine in 1/10 epochs with and without XLA. The memory had shown 500 megabytes decrease in consumption, a 14% decrease in CPU utilization, and a 5% decrease in disk memory. This means that there is a slight improvement in the running status of the machine in the first run in 10 epochs.

Table 2. Decrease Machine's Running Status in 1/10 Epochs

Epochs	Memory (in Gb)	CPU	Disk memory
1/10	0.5	14%	5%

Table 3 shows the difference between the running status of the machine in 10/10 epochs with and without XLA. The memory had shown a 200-megabyte decrease in memory usage. There is also a 5% decrease in disk memory however there is a 16% decrease in CPU utilization when the training is finished when XLA is utilized. Keeping the slight improvement in the machine status

Table 3. Decrease Machine's Running Status in 10/10 Epochs

Epochs	Memory (in Gb)	CPU	Disk memory
10/10	0.2	16%	5%

Table 4 shows the difference between the running status of the machine in 1/50 epochs with and without XLA. The memory had shown 400 megabytes decrease in memory and 79% decrease in disk memory and a 95% decrease in CPU utilization when the training finished when XLA is utilized. Meaning that there is still a slight improvement in the start of the training.

Table 4. The Machine's Running Status in 1/50 Epochs

Epochs	Memory (in Gb)	CPU	Disk memory
1/50	0.4	7%	79%

Table 5 shows the difference between the running status of the machine in 50/50 epochs with and without XLA. The disk memory had shown an 80% decrease in memory however there is a 100 megabytes increase in memory is a 3% increase in CPU utilization

when the training is finished when XLA is utilized. This might mean that due to the increased span of training the machine increased its need for resources however due to the XLA the increase is not that significant and is still able to keep a large decrease in disk memory.

Table 5. The Machine's Running Status in 50/50 Epochs

Epochs	Memory (in Gb)	CPU	Disk memory
50/50	-.10	-3%	80%

Table 6 shows the difference between the total running time of training LSTM with and without XLA. There is a 171.799-second increase when training using XLA.

Table 6. The Running Time Difference Between with XLA and without XLA in 10 epochs

Total Seconds w/o XLA	Total Seconds w/ XLA	Total Second Difference	Total Minute Difference
2035.223	2207.002	-171.799	-2.86

Table 7 shows the difference between the total running time of training LSTM with and without XLA. There is a 536-second decrease when training using XLA. Meaning that there is a greater decrease in training time the more epoch is utilized.

Table 7. The Running Time Difference Between with XLA and without XLA in 50 epochs

Total Seconds w/o XLA	Total Seconds w/ XLA	Total Second Difference	Total Minute Difference
11710.7507	11174.4598	536.2909	8.938181667

DISCUSSION

Using XLA training an LSTM model had shown a slight improvement in the machine status especially in disk memory and its training time. There is a consistent slight improvement in the running status when XLA is used on LSTM in 10 epochs having a 6% decrease in CPU utilization, 500 megabytes decrease in memory consumption, and 5% decrease in disk memory at the start, and 16% decrease in CPU utilization, 100 megabytes increase in the memory consumption and 5% decrease in disk memory, however, there is a 2-minute increase in training time of the model. The model on the other hand had shown a more promising result in training time reducing it at least 8 minutes and 55 seconds and is still able to present slight improvement at the start of the training with 400 megabytes decrease in consumption, 7% decrease in CPU utilization, and 79% decrease in disk memory. However, the results show 100 megabytes increase in memory consumption and a 3% increase in CPU utilization though it is not that significant.

CONCLUSIONS AND RECOMMENDATIONS

The research had concluded that XLA can provide a slight improvement in the training of the model in terms of its memory usage, CPU utilization, training time and its effect is much more significant when used in a larger training time. However, due to the lack of time, the researchers have not tested XLA with pruning and effect in combination with XLA.

REFERENCES

A beginner's introduction to Natural Language Processing (NLP). (2021). Retrieved November 11, 2021, from https://www.cloudmoyo.com/blog/ai-ml-automation/a-beginners-introduction-to-natural-language-processing/

Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020). A smart chatbot architecture-based NLP and Machine Learning for Health Care Assistants. *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*. doi:10.1145/3386723.3387897

Chadha, P., & Siddagangaiah, T. (n.d.). *Performance Analysis of Accelerated Linear Algebra Compiler for TensorFlow*. Retrieved November 15, 2021, from http://parthchadha.github.io/xla_report.pdf

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*(10), 2222-2232. doi:10.1109/tnnls.2016.2582924

Hou, L., Zhu, J., Kwok, J. T., Gao, F., Qin, T., & Liu, T. (2019). *Normalization Helps Training of Quantized LSTM*. Retrieved November 13, 2021, from https://openreview.net/forum?id=rJgB34rx8r

Husak, V., Lozynska, O., Karpov, I., Peleshchak, I., Chirun, S., & Vysotskyi, A. (n.d.). *Information system for recommendation list formation of clothes style image selection according to user's needs based on NLP and chatbots*. Retrieved November 11, 2021, from http://ceur-ws.org/Vol-2604/paper54.pdf

Klie, L. (2014). Neural Networks Reach into Virtual Assistants. Published. https://www.destinationcrm.com/Articles/ReadArticle.aspx?ArticleID=98749#:~:text=Neural%20Networks%20Reach%20into%20Virtual%20Assistants%20Neural%20networking%2C,has%20placed%20new%20attention%20on%20the%20decades-old%20technology

Lazar, I. (2021). *Quantifying the benefit of intelligent virtual assistants to improve meeting experiences AI-powered virtual assistants reduce costs and improve productivity*. Retrieved from https://www.cisco.com/c/dam/en/us/solutions/collateral/collaboration/the-benefit-of-intelligent-virtual-assistants.pdf?ccid=

Liddy, E. D. (2001). *Natural Language Processing*. Retrieved November 12, 2021, from https://surface.syr.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1019&context=cnlp

Lyu, Q., & Zhu, J. (n.d.). *Revisit long short-term memory: An optimization perspective.* Retrieved November 13, 2021, from http://ml.cs.tsinghua.edu.cn/~jun/pub/lstm-parallel.pdf

McGregor, M. (2020). *Google BERT NLP Machine Learning Tutorial.* FreeCodeCamp. https://www.freecodecamp.org/news/google-bert-nlp-machine-learning-tutorial

Mills, T. (2018). *What Is Natural Language Processing and What Is It Used For*? Forbes. Retrieved from https://www.forbes.com/sites/forbestechcouncil/2018/07/02/what-is-natural-language-processing-and-what-is-it-used-for/?sh=d8b6af05d71f

Mirwan, Nugroho, A., Hendarta, F., Hidayatillah, R., Hassan, F., & Nana, K. P. (2018). Virtual assistant using LSTM networks in Indonesian. *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. doi:10.1109/isriti.2018.8864448

Molnar, G., & Szuts, Z. (2018). The role of Chatbots informal education. *2018 IEEE 16th* International *Symposium on Intelligent Systems and Informatics (SISY)*. doi:10.1109/sisy.2018.8524609

Nan, G., Wang, C., Liu, W., & Lombardi, F. (2020). DC-LSTM: Deep compressed LSTM with low bit-width and structured matrices. *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. doi:10.1109/iscas45731.2020.9180869

Patil, S., Mudaliar, V. M., Kamat, P., & Gite, S. (2020). LSTM based Ensemble Network to enhance the learning of long-term dependencies in chatbot. *International Journal for Simulation and Multidisciplinary Design Optimization, 11*, 25. https://doi.org/10.1051/smdo/2020019

Silfa, F., Dot, G., Arnau, J., & Gonzàlez, A. (2018). E-PUR: An Energy-Efficient Processing Unit for Recurrent Neural Networks. In *Proceedings of the 27th International Conference on* Parallel *Architectures and Compilation Techniques*. doi:10.1145/3243176.3243184

Singh, R. (2020). *Accelerate your training and inference running on Tensorflow*. Medium. Retrieved from https://towardsdatascience.com/accelerate-your-training-and-inference-running-on-tensorflow-896aa963aa70

Sri, M. (2021). NLP in Virtual Assistants. *Practical Natural Language Processing with Python,* 185-199. doi:10.1007/978-1-4842-6246-7

Tiled layout | XLA |. (2021). *TensorFlow*. Retrieved December 12, 2021, from https://www.tensorflow.org/xla/tiled_layout

Van der Westhuizen, J., & Lasenby, J. (2018). *The unreasonable effectiveness of the Forget Gate*. Retrieved November 11, 2021, from https://arxiv.org/abs/1804.04849

Wang, M., Wang, Z., Lu, J., Lin, J., & Wang, Z. (2019). E-LSTM: An efficient hardware architecture for long short-term memory. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* 9(2), 280-291. doi:10.1109/jetcas.2019.2911739

Wang, S., Li, Z., Ding, C., Yuan, B., Qiu, Q., Wang, Y., & Liang, Y. (2018). C-LSTM. *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays,* 11-20. doi:10.1145/3174243.3174253

What Does Intelligent Virtual Assistant Mean? (2021). Techopedia. Retrieved from https://www.techopedia.com/definition/31383/intelligent-virtual-assistant

What is Natural Language Processing? (2020). Ibm.Com. Retrieved November 30, 2021, fro`m: https://www.ibm.com/ cloud / learn / natural-languageprocessing # :~: text= What%20is%20natural%20language%20processing%3F%20Natural%20language%20pr ocessing,in%20much%20the%20same%20way%20human%20beings%20can.