Milk Production Prediction Model with LSTM

Choo Tze Yuet
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp060673@mail.apu.edu.my

Heng Yuen Zi
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp061096@mail.apu.edu.my

Khaw Zhi Pei
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp060846@mail.apu.edu.my

Sarveashwaran A/L Pathmanathan
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp060637@mail.apu.edu.my

Zailan Arabee Abdul Salam
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
zailan@apu.edu.my

***Abstract*—The quality of milk provided to household livelihood is highly dependent on the quality of raw milk from the milk production. It is crucial to control the quality of raw milk to maintain the food security and nutritious value for household livelihood. This paper proposes a milk production prediction model by using Long short-term memory (LSTM) techniques.**

Keywords—Long short-term memory (LSTM), milk production prediction, time series data, deep learning, activation function

I. Introduction

Milk is one of the most nutritious drink in the world. Recently, milk production has increased drastically and this activity can drive the economy of the whole world. It is important to predict the milk production in order to cope with the demand of the market while at the same time maintaining the freshness and nutritious value of the milk.

(Punyapornwithaya et al., 2021) stated that Seasonal Autoregressive Integrated Moving Average (SARIMA), Exponential Smoothing (ETS) and SARIMA-ETS is used in milk production prediction modelling. The Root Mean Square Error (RMSE) value for SARIMA, ETS and SARIMA-ETS is 652.64, 458.40 and 467.71. It is clearly seen that ETS model is having is giving the best accuracy out of these three model compared in this paper. In this paper we will be using LSTM for our prediction model as LSTM as the milk production is highly dependent on time series and LSTM is best for handling the time sequence data input. Our model produce better accuracy with RMSE value of 19.0688.

II. Literature Review

A. Similar Projects

There are a few researches regarding LSTM algorithms carried out in the past few years. These papers will be discussed here to understand the strength of LSTM algorithm.

(Caihong et al., 2018) stated that LSTM and ANN are compared to observe the performance in predicting rainfall-runoff to prevent flood. Although both models show a good result in term of non-linearity, LSTM is a better choice in term of intelligence and stability. LSTM has higher correlation efficient than ANN showing that LSTM can adapt to various changes, particularly peak discharge while ANN is only subjected to rainfall. Various parameters are taken to compare the performance of the two models such as different periods and lead time. LSTM appeared to have higher R^2 value than ANN in calibration period with 86 flood event series and validation period with 12 flood event series.

As conducted in research paper (Tianyu et al., 2019), LSTM is used to predict the flash flood. LSTM is proved to address the weakness of RNN in term of continuous dependencies. LSTM is capable to keep track with the time-series changes with its strength in shortening the time taken and memory storage. Due to the special memory cells LSTM has, LSTM is proved to perform better in dealing with time-series data.

Murphy et al. (2014) had explained how milk production is forecasted by Artificial Neural Network (ANN) instead of LSTM algorithm. Back propagation algorithm is adopted by ANN that reduces the error rate which involves the evaluation between net output and expected output. Subsequently, it is passed to the network and alter the synaptic weight in order to eliminate chance of error. ANN is compared with MLR in this case and ANN outperforms MLR. ANN is compared with MLR in this case and ANN outperforms MLR.

B. Methodology/ Approach

In this paper, the datasets that we used is from GitHub. The first column of the dataset is the date recorded, in year-month format and the second column of the dataset is the milk production record for the month of the year.

In this paper, we applied the deep learning techniques of LSTM that is highly dependent on the time series data. LSTM is widely used for model that highly relies on the time sequence of data where the result will alter as the time sequence data alter. LSTM is used in this paper as it is known by having forget gates that is able to throw the unused data out of the model and add new input to it at the same time.

C. Conclusion/ Recommendation

With this algorithm, we are able to predict the milk production accordingly. Our output gives the milk production prediction for the following month of the year. Section 2 provides the source of our dataset and explains the algorithm of our model. Section 3 presents the result from the prediction model and discuss about our result. Section 4 concludes the result of our model.

III. Materails and Methods

A. Source of material

The set of data and source code for this study was found and obtained from a website named GitHub. GitHub is a code facilitating stage for variant control and cooperation. It allows people from any corner of the world to work on certain projects together. Studies say that over 83 million developers are contributing together via GitHub for growth of software according to GitHub Docs. Hence, a project of prediction for milk production was obtained from this website for research in concern of content quality which would be a great advantage for experimental understanding. In this project, the algorithm is built using Python programming language and the dataset provided is of a milk producing organisation's monthly milk production from 1962 to 1975. There are a total of 168 rows of data provided to conduct the experiment. Each row consists of the year with month and production count.

B. Device

For this research, a Huawei laptop was used for the whole time of the study. Table I shows the designation of the gadget used.

TABLE I. Specification of Huawei Laptop

Specification	Description
Model Name	LAPTOP-861VBTC9
Processor	AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz
Memory	16.0 GB
Storage	Hard Disk Drive (HDD) – 2 TB Solid State Drive (SSD) – 222 GB
Operating System (OS)	Windows 11 Home

C. Software

Google Colab was utilized all through the study. Google Colab, also known as Collaboratory is a product of Google Research. Google Colab is truly appropriate for this research since it can test artificial intelligence based models and accessible online without requiring any downloading which makes it extremely simple and advantageous to be utilized. Not just that, results of the execution of the algorithm is saved in Google Drive readily. This makes it simpler to impart the outcomes with other researchers.

IV. Algorithm Implementation

A. Introduction to LSTM

This research is done using a machine learning algorithm called Long Short-Term Memory algorithm (LSTM). LSTM is a type of Recurrent Neural Network (RNN) that works well with sequential data. It was introduced by Sepp Hochreiter and Jurgen Schmidhuber in 1997 (Yanhui, 2021). The main reason LSTM was invented is to solve the Vanishing Gradient Problem of RNN. The Vanishing Gradient Problem has a connection with gradient descent algorithms. So it is only sensible to develop the idea from it. Fig 1. Shows how a typical gradient descent algorithm looks like.

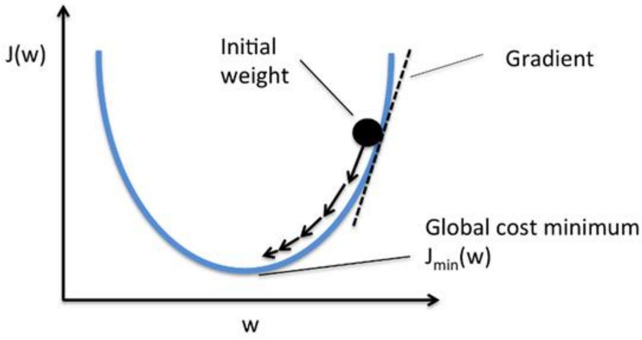

Fig. 1. Gradient Descent Algorithm

This algorithm is merged with a backpropagation algorithm to recondition the synapse weights all over the neural network. RNN acts a little different due to the hidden layer of an observation is being utilized to exercise the next observation's hidden layer, also it must be known that the cost function of the neural network is computed for each observation present in given data set. Coming back to the issue addressed, Vanishing Gradient Problem happens when the algorithm of backpropagation locomotes back through every neuron of the network to refresh the weights. RNN's nature is that cost function calculated at inner layer of the neural network will contribute to alter the weights of shallower layer neurons. This alteration is computed by the nature of multiplication, where the gradient computed in a stage that is deep in the neural network will be multiplied back through prior weights in the network. Putting in simpler words, the gradient computed inner in the neural net is weakened as it locomotes back by the network. This causes the gradient to disappear or more accurately, Vanish (McCullum, 2022).

1) Working Principle of LSTM

Before understanding how LSTM solves the issue of vanishing gradient, it is important to know how the LSTM model works. The LSTM is comprised of four neural nets with various memory blocks recognized as cells. A regular LSTM unit comprises of a cell, a forget gate, an input gate and an output gate. The gates can be considered as filters. Each gate is also their own neural network according to Dolphin (2020). The progression of data in and out of the cell is constrained by the three gates mentioned, they manipulate the models' memory and the cell keeps memory of the values throughout inconsistent time spans.

The three gates mentioned earlier are also the entrances for data to come into the model. The input gate figures out which input value ought to be utilized to update the memory. The sigmoid function decides if to permit 0 or 1 through whereas the tanh function doles out weight to the provided data, deciding their significance on a scale of - 1 to 1. Forget gate identifies the subtleties that ought to be eliminated from the block. It is chosen by a sigmoid function in the forget gate. For each number in the cell state, it takes a gander at the prior state and the content input then generates a number between 0 and 1. Lastly for the output gate, The block's input and memory are used to determine the output. The sigmoid decides if to permit 0 or 1 values through while tanh figures out which values are permitted to go through 0, 1. The tanh function also allocates weight to the given values, deciding their pertinence on a size of - 1 to 1 and multiply it with the sigmoid result.

2) LSTM Network

RNNs are made up of an order of repeating neural network modules. Similarly LSTM is also built up that way, its repeating module consists of four layers that communicate with each other as shown in Fig. 2.

The horizontal line at the highest point of Fig 2. is the cell state of LSTM. There are just a couple of direct interactions down the chain and it is simple for information to go down the chain without being changed. LSTM can remove or add data to the cell state, which is constrained by the gates. The sigmoid layer produces numbers going from zero to one as referenced before, specifying the amount of every component that could be permitted to pass. Zero represents none ought to be permitted and one represents all ought to be permitted. The input, output and forget gate's main purpose is to control and protect the cell state.

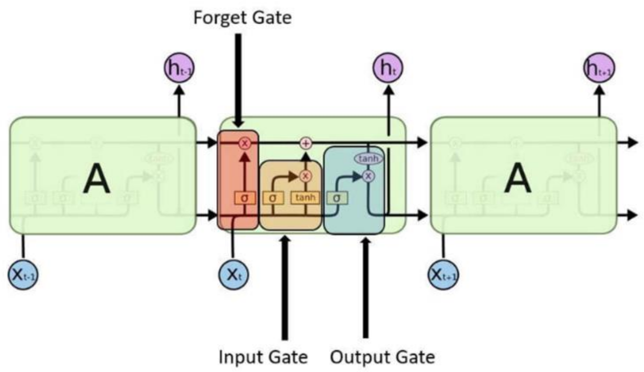

Fig. 2. Layers of recurrent unit of LSTM model

3) Cycle of LSTM

- Data to be removed is determined from the earlier step utilizing the forget gate.
- New data is looked up to update the cell state with help of tanh function and input gate.
- Cell state is updated with the data produced by the previous two steps.
- Output gate and the squashing operation gives beneficial data (Kalita, 2022).

4) How LSTM solves the Vanishing Gradient Problem

It uses a special additive gradient structure that incorporates direct admittance to the forget gate's initiations, empowering the network to energize wanted conduct from the error gradient utilizing continuous gates update on each time step of the learning procedure (Arbel, 2018).

B. Purpose

LSTM is mainly used for predictions and classification. Hence, it is used to predict milk production for future using given data of 168 months which is equivalent to 14 years. LSTM produces predictions with high accuracy because it is considered to be the strongest among all RNN to perform forecasting principally when having a longer-term trend in the dataset. This is due to LSTM cells adding long haul memory in a much more performant way since it permits more parameters to accomplish learning (Keith, 2022). When talking about the parameters, the following part will be discussing about the parameters set in the study and how changing them affects the final results.

C. Parameters

1) Batch size

Batch size helps in characterising the quantity of sample data to be trained as one set. It is declared using the n_input variable in the source code. For instance, in the research there is an example where the last 12 rows of data are chosen and milk productions of 3 months have been used as one batch to understand the working principle of LSTM. So 12 rows of data will make 4 batches with 3 months' milk production in each batch. The LSTM model is trained using these batches just that with different batch sizes according to dataset and needs . Notwithstanding, there is a constraint where size of a batch should be more than or equivalent to one and less than or equivalent to the sample size. It give advantageous in preparing the information in bunch as opposed to individually.

2) Activation Function

Activation function is the method in an artificial neuron that conveys a result in light of inputs. These functions in neurons are a significant piece of the job that the neurons play in the latest artificial neural networks (Techopedia, 2018). One method for understanding the initiation capability is to take a gander at a visual "model" of the fake neuron. The enactment capability is toward the "end" of the brain structure and compares generally to the axon of a natural neuron. There are many types of activation functions such as Sigmoid, tanh, Softmax, ReLU and more. In this research Rectified Linear Unit (ReLU) function is used. The ReLU is the most involved enactment capability on the planet right now. Since, it is utilized in practically all the convolutional brain organizations or profound learning. The capability and its subsidiary both are monotonic as stated by Sharma (2018).

V. Results and Discussion

A. Number of input

TABLE II. Different of RMSE When Number of Input is 9 Months, 12 Months and 15 Months

Test	RMSE		
	9 Months	***12 Months***	***15 Months***
1	48.0607	26.7269	20.5836
2	38.1537	26.3045	18.1698
3	36.5289	20.4823	17.3745
4	39.0365	19.3910	20.1734
5	30.8975	27.7421	21.5931
6	39.6340	19.8945	16.5106
7	49.0461	25.0888	18.8947
8	29.5214	22.2456	22.3101
9	38.2406	22.8453	17.6174
10	28.5589	23.2659	17.4612
Average	36.7678	23.3987	19.0688

The model is tested with different number of input which are 9 months, 12 months and 15 months. As shown in Table II, RMSE decreases when number of input increases. When the number of input is increased to 15 months, the average of RMSE is 19.0688. Therefore, it can be concluded that a higher number of input leads to a lower RMSE. However, the number of input is limited to 168 months.

B. Activation Functions

Activation functions are modified by using ReLU, tanh and sigmoid. In Table III, sigmoid has a result of 137.8138 as the average RMSE while ReLU has a value of 23.3987 as the average RMSE. It is obvious that there is a huge difference between sigmoid and ReLU activation functions' result. A conclusion can be made that ReLU is the most suitable activation function to be used in this model.

TABLE III. DIFFERENT OF RMSE BASED ON RELU TANH AND SIGMOID

Test	**RMSE**		
	ReLU	**Tanh**	**Sigmoid**
1	26.7269	31.8173	213.0956
2	26.3045	32.1883	67.0943
3	20.4823	31.3463	213.1466
4	19.3910	37.3746	108.6010
5	27.7421	29.2750	140.3097
6	19.8945	26.1410	195.3923
7	25.0888	41.0016	65.1971
8	22.2456	32.9564	165.6004
9	22.8453	27.4063	145.9994
10	23.2659	35.2807	63.7018
Average	23.3987	32.4788	137.8138

C. Comparison between Default LSTM and Modified LSTM

The number of input is modified to compare the changes in graph. However, the activation function is fixed which both models use ReLU activation function.

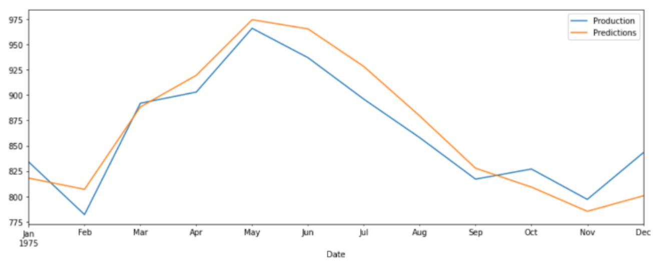

Fig. 3. Default LSTM, using 12 months number of input with ReLU activation function

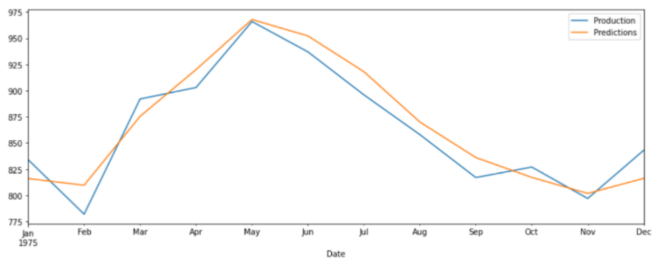

Fig. 4. Modified LSTM, using 15 months number of input with ReLU activation function

The prediction line is nearer to production line in graph of Fig. 4 as compared to Fig. 3. In Fig. 3, the interception points between production line and prediction are 4 while Fig. 4 has 7 interception points. The closer the distance between both lines, the stronger the correlation, the more accurate the model's prediction. Hence, it can be concluded that LSTM works effectively when receiving more number of input that can increase the accuracy.

VI. CONCLUSIONS

In this paper, we are finding the optimal performance of the LSTM model by modifying the parameters which are the number of input and activation function to increase the accuracy of milk production prediction. Based on our findings, when increasing the number of input, the accuracy of prediction results will also increase. More data can be used for training to study the past behaviours of patterns and trends. However, it was found that 15 months of input training data will obtain an optimal result. When using more than 15 months of input, it did not get better results because the given dataset is limited to 168 months. We also found that ReLU is the optimal activation function for this LSTM model and will get the most accurate result.

The default LSTM get an average of 23.3987 RSME value while modified LSTM get 19.0688 RSME value. The prediction results have increased by 18.5% after modified the parameters in LSTM model. The limitation of this study is there are only two parameters that can be modified which are the number of input and activation function. To improve the study, we can add more parameters such as number of layers and nodes so that better optimization can be achieved.

REFERENCES

Punyapornwithaya, V., Jampachaisri, K., Klaharn, K. & Sansamur, C. (2021, November 30). Forecasting of Milk Production in Northern Thailand Using Seasonal Autoregressive Integrated Moving Average, Error Trend Seasonality, and Hybrid Models. *Journal of Veterinary Epidemiology and Economics*, 8. https://doi.org/10.3389/fvets.2021.775114

Caihong, H., Qiang, W., Hui, Li., Shengqi, J., Nan, L., & Zhengzheng, L. (2018, October 30). Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation. *Water*, 10(11), 1543. https://doi.org/10.3390/w10111543

Tianyu, S., Wei, D., Jian, W., Haixing, L., Huicheng, Z., & Jinggang, C. (2019, December 21). Flash Flood Forecasting Based on Long Short-Term Memory Networks. *Water*, 12(1), 109. https://doi.org/10.3390/w12010109

Murphy, M. D., O'Mahony, M. J., Shallo, L., French, P., & Upton, J. (2014, June 1). Comparison of modelling techniques for milk-production forecasting. *Journal of Dairy Science*, 97(6), 3352–3363. https://doi.org/10.3168/jds.2013-7451

Hello World. (n.d.). Retrieved from GitHub Docs: https://docs.github.com/en/get-started/quickstart/hello-world

Yanhui, C. (2021, March 8). A Battle Against Amnesia: A Brief History and Introduction of Recurrent Neural Networks. Retrieved from Towards Data Science: https://towardsdatascience.com/a-battle-against-amnesia-a-brief-history-and-introduction-of-recurrent-neural-networks-50496aae6740

McCullum, N. (2022). The Vanishing Gradient Problem in Recurrent Neural Networks. Retrieved from nickmccullum: https://www.nickmccullum.com/python-deep-learning/vanishing-gradient-problem/

Dolphin, R. (2020, October 22). LSTM Networks | A Detailed Explanation. Retrieved from Towards Data Science: https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9#:~:text=LSTMs%20use%20a%20series%20of,each%20their%20own%20neural%20network.

Kalita, D. (2022, March 22). An Overview on Long Short Term Memory (LSTM). Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2022/03/an-overview-on-long-short-term-memory-lstm/#:~:text=It%20is%20used%20for%20time,processing%2C%20prediction%2C%20and%20classification.&text=LSTM%20has%20feedback%20connections%2C%20unlike,such%20as%20speech%20or

Arbel, N. (2018, December 21). How LSTM networks solve the problem of vanishing gradients. Retrieved from Data Driven Investor: https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577#:~:text=LSTMs%20solve%20the%20problem%20using,step%20of%20the%20learning%20process.

Keith, M. (2022, August 5). Exploring the LSTM Neural Network Model for Time Series. Retrieved from Towards Data Science: https://towardsdatascience.com/exploring-the-lstm-neural-network-model-for-time-series-8b7685aa8cf

Activation Function. (2018, June 1). Retrieved from techopedia: https://www.techopedia.com/definition/33261/activation-function

Sharma, S. (2017, September 6). Activation Functions in Neural Networks. Retrieved from Towards Data Science: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6