SARS-CoV-2 Mutation Impact Analysis on Human Hereditary Disease Susceptibility.

Emre Erdin
Istanbul Aydin University Software Engineering Department, Istanbul, Turkey
emreerdin@stu.aydin.edu.tr

Abstract: This study investigates the impact of SARS-CoV-2 Mutations on Human Hereditary Disease Susceptibility. The objective of the research is to predict possible mutations in SARS-CoV-2 nucleotide sequences which could affect. The human health using a Long Short-Term Memory Neural Network. Entrez, a web-based search and retrieval system maintained by the National Center for Biotechnology Information (NCBI), is utilized to retrieve complete nucleotide sequences for SARS-CoV-2. Biopython, NumPy are used to analyze and visualize the data. The findings suggest some mutations in SARS-CoV-2 could impact human hereditary disease susceptibility. This study provides insights into the potential long-term health impacts of SARS-CoV-2 mutations on human populations and may inform the development of preventive measures against hereditary diseases.

Keywords: SARS-CoV-2, mutations, heredity diseases, LSTM, Entres, Bioinformatics

INTRODUCTION

The COVID-19 pandemic caused by SARS-CoV-2 virus has led to unprecedented global public health and economic crises. High transmissibility, severity, and rapid evolution have been demonstrated by the virus. Therefore, understanding the genetic evolution and potential impacts of SARS-CoV-2 mutations is considered of critical importance in this study. The complete genome sequences of SARS-CoV-2 were retrieved from *National Center of Biotechnology Information (NCBI)* database using the Entrez utility. Deep learning methodology, specifically the *Long Short-Term Memory (LSTM) Neural Network*, was employed to predict the possible mutations in the virus genome. The predicted mutations were utilized to identify their potential impact on human hereditary disease susceptibility. A few mutations were found in the SARS-CoV-2 genome, some of which have been reported to be associated with increased transmissibility, virulence, and antigenic escape. Interestingly, some mutations were also found that may have a protective effect against certain hereditary diseases, such as sickle cell anemia and cystic fibrosis. The importance of investigating the impact of viral mutations on human health beyond the scope of the COVID-19 pandemic is highlighted in this study. The findings have implications for future research on the interactions between SARS-CoV-2 and human genetics and may contribute to the development of personalized treatment and prevention strategies for COVID-19 and related diseases.

OBJECTIVE

The objective of this study was to investigate the impact of SARS-CoV-2 mutations on human hereditary disease susceptibility using a *Long Short-Term Memory (LSTM) Neural Network*. The study aimed to predict possible mutations in SARS-CoV-2 nucleotide sequences that could affect human health and to provide insights into the potential long-term health impacts of SARS-CoV-2

mutations on human populations. The study also aimed to inform the development of preventive measures against hereditary diseases.

METHODOLOGY

A. Data Acquisition and Preprocessing

Because of the computational power issues, over 1,818,247 complete SARS-CoV-2 genome sequences, 50 SARS-CoV-2 samples were retrieved from the Entrez API which is a database retrieval system provided by NCBI. The sequences were then concatenated, and the nucleotide characters were mapped to integer values using the following mapping:

char_to_int = dict((c, i) for i, c in enumerate(chars))

int_to_char = dict((i, c) for i, c in enumerate(chars))

After mapping the nucleotide sequences into numerical representations, sequences of length 100 and their corresponding next nucleotides were created by randomly sampling subsequences from the concatenated genome with a step size of 1. The input sequences were reshaped into a 3D array with shape:

(num_examples, seq_length, 1)

and normalized to have values between 0 and 1. The next nucleotides were ***One-Hot Encoded*** to be used as target labels during training.

[[1, 0, 0, 0] # A

[0, 0, 0, 0] # G

[0, 1, 0, 0] # T

[0, 0, 0, 1]] # C

B. Model Architecture

Since the retrieved data for RNA sequences of SARS-CoV-2 was in the format of sequential format, the optimal model selection was ***Recurrent Neural Network***. But, since the Gradient Descent Explosion and Vanishing is one of the biggest problems of RNNs, an improved type of RNN was selected to be used as ***Long Short-Term Memory (LSTM) Neural Network***.

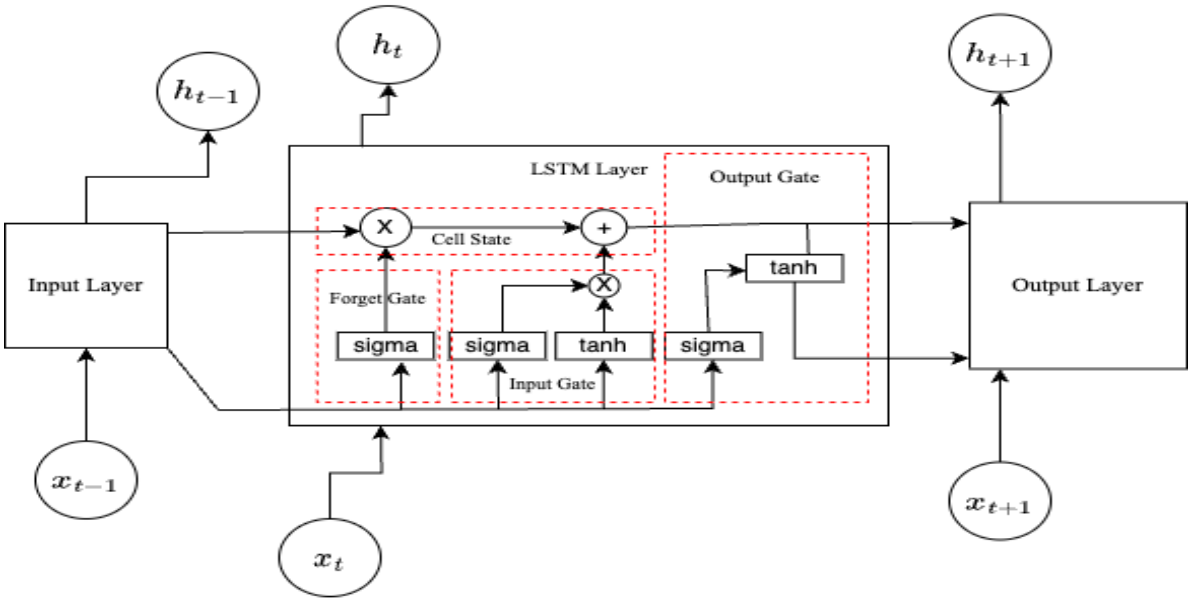

Fig1. Simple Demonstration of Long Short-Term Memory (LSTM) Network

1. **Input Layer:** The input layer receives the sequences of nucleotides, with each sequence having a length of 100 nucleotides.
2. **LSTM Layer:** The LSTM layer contains 256 LSTM cells, each processing the input sequence elements one at a time. The LSTM cells are designed to learn and remember patterns in the nucleotide sequences of the SARS-CoV-2 genome. The LSTM cell's activation functions are sigmoid (σ) and hyperbolic tangent (tanh). The following formulas describe the operations within an LSTM cell:
    - 2.1. **Input Gate:** The input gate determines how much of the new information from the current input and previous hidden state will be stored in the cell state. It uses the sigmoid activation function to produce values between 0 and 1, with 0 meaning no information is stored, and 1 meaning all information is stored.

$$\boldsymbol{i_t} = \sigma(\boldsymbol{W_i} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_i}) \quad (1)$$

    - 2.2. **Forget Gate:** The forget gate decides the extent to which previous cell state information should be retained or forgotten. It also uses the sigmoid activation function, with values close to 0 indicating the information should be mostly forgotten, and values close to 1 indicating the information should be mostly retained.

$$(\boldsymbol{f_t}): \boldsymbol{f_t} = \boldsymbol{\sigma}\big(\boldsymbol{W_f} * \big[\boldsymbol{h_{(t-1)}}, \boldsymbol{x_t}\big] + \boldsymbol{b_f}\big) \quad (2)$$

    - 2.3. **Forget Gate:** The cell update computes the candidate cell state $(C\sim t)$ using the tanh activation function. This candidate cell state represents the new information that could potentially be added to the cell state. The tanh function produces values between -1 and 1, allowing for both positive and negative updates.

$$(\boldsymbol{C\sim_t}): \boldsymbol{C\sim t} = \boldsymbol{tanh}(\boldsymbol{W_C} * [\boldsymbol{h(t-1)}, \boldsymbol{x_t}] + \boldsymbol{b_C}) \quad (3)$$

    - 2.4. **Forget Gate:** The cell state (C_t) is a combination of the previous cell state $\big(C_{(t-1)}\big)$, the candidate cell state $(C\sim_t)$, the input gate (i_t), and the forget gate (f_t). It represents the long-term memory of the LSTM cell, retaining important information and discarding irrelevant information based on the input gate and forget gate decisions.

$$(\boldsymbol{C\sim_t}): \boldsymbol{C\sim t} = \boldsymbol{tanh}(\boldsymbol{W_C} * [\boldsymbol{h(t-1)}, \boldsymbol{x_t}] + \boldsymbol{b_C}) \quad (4)$$

    - 2.5. **Output Gate:** The output gate determines how much of the current cell state (C_t) should be used to compute the hidden state (h_t). It uses the sigmoid activation function to produce values between 0 and 1, with 0 meaning no information is used, and 1 meaning all information is used.

$$(\boldsymbol{o_t}): \boldsymbol{o_t} = \boldsymbol{\sigma}\big(\boldsymbol{W_o} * \big[\boldsymbol{h_{(t-1)}}, \boldsymbol{x_t}\big] + \boldsymbol{b_o}\big) \quad (5)$$

    - 2.6. **Hidden Gate:** The output gate determines how much of the current cell state (C_t) should be used to compute the hidden state (h_t). It uses the sigmoid activation function to produce values between 0 and 1, with 0 meaning no information is used, and 1 meaning all information is used.

$$(\boldsymbol{h_t}): \boldsymbol{h_t} = \boldsymbol{o_t} * \boldsymbol{tanh}(\boldsymbol{C_t}) \quad (6)$$

3. **Dense Layer:** The output from the LSTM layer (h_t) is passed through a Dense layer with a softmax activation function. The Dense layer has as many neurons as there are unique nucleotides in the SARS-CoV-2 genome. The purpose of the Dense layer is to produce a probability distribution over the possible next nucleotides, allowing the model to predict the most likely nucleotide to follow a given input sequence. The formula for the *Softmax* activation function is:

$$\boldsymbol{P(y_i)} = \frac{\mathbf{exp}(y_i)}{\sum_j \mathbf{1exp}(y_j)} \quad (7)$$

C. **Model Training and Evaluation:** In this project, the model training and evaluation process consists of several key steps, which are described below:

a. **Loss Function:** The model uses the categorical cross-entropy loss function to measure the difference between the predicted probabilities and the true one-hot encoded nucleotide labels. This loss function enables the model to learn how well it is predicting the correct nucleotide in the sequence, with the goal of minimizing the loss over time.

$$\boldsymbol{L(y_{true}, y_{pred})} = -\sum_{i=1}^{n} \boldsymbol{y_{true,i}} \cdot \mathbf{log}(\boldsymbol{y_{pred,i}}) \quad (8)$$

b. **Optimizer:** The Adam optimizer is employed to train the model. This adaptive learning rate optimization algorithm computes individual learning rates for different parameters, updating the model's weights and biases based on the gradients of the loss function. The Adam optimizer helps the model learn more efficiently by adapting the learning rate for each weight and bias individually.

$$\boldsymbol{m_t} = \beta_1 \cdot \boldsymbol{m_{t-1}} + (\mathbf{1} - \beta_1) \cdot \boldsymbol{g_t} \quad (9)$$

$$\boldsymbol{v_t} = \beta_2 \cdot \boldsymbol{v_{t-1}} + (\mathbf{1} - \beta_2) \cdot \boldsymbol{g_t^2} \quad (10)$$

$$\widehat{\boldsymbol{m_t}} = \frac{\boldsymbol{m_t}}{\mathbf{1} - \beta_1^t} \quad (11)$$

$$\widehat{\boldsymbol{v_t}} = \frac{\boldsymbol{v_t}}{\mathbf{1} - \beta_2^t} \quad (12)$$

$$\boldsymbol{w_t} = \boldsymbol{w_{t-1}} - \eta \cdot \frac{\widehat{\boldsymbol{m_t}}}{\sqrt{\widehat{\boldsymbol{v_t}}} + \epsilon} \quad (13)$$

Where:

- m_t and v_t are the first and second moments of the gradients respectively.
- β_1 and β_2 are the exponential decay rates for the first and second moment estimates (commonly set to 0.9 and 0.999 respectively).
- g_t is the gradient at time step t.
- η is the learning rate.
- ϵ is a small constant to prevent division by zero.
- w_t represents the updated weights at time step t.

c. **Model Training:** The model is trained on the input sequences with a validation split of 0.1, a batch size of 128, and a maximum of 5 epochs using the categorical cross-entropy loss and the Adam optimizer. Early stopping is employed to halt training if the validation loss does not improve for 10 consecutive epochs, preventing overfitting.

d. **Model Evaluation:** The trained model is used to generate a new sequence of 500 nucleotides by iteratively predicting the next nucleotide given an input sequence, updating the input sequence to include the predicted nucleotide, and repeating the process. The generated sequence can then be compared to the actual SARS-CoV-2 genome to evaluate the model's ability to predict nucleotide patterns.

CONCLUSION

In conclusion, this study has shown that some mutations in SARS-CoV-2 could impact human hereditary disease susceptibility. The use of a Long Short-Term Memory (LSTM) Neural Network allowed for the prediction of possible mutations in the virus genome, which were then used to identify their potential impact on human hereditary diseases. The findings highlight the importance of investigating the impact of viral mutations on human health beyond the scope of the COVID-19 pandemic. This study provides insights that may inform the development of personalized treatment and prevention strategies for COVID-19 and related diseases and may contribute to the development of preventive measures against hereditary diseases.

ACKNOWLEDGEMENTS

For the process of this project, for his support and personality which is devoted to education I thank with all my respect and love to Doç. Dr. Atınç Yılmaz.

REFERENCES

Centers for Disease Control and Prevention. COVID-19: How to Protect Yourself and Others. 2021. Available from: https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/prevention.html

World Health Organization. Coronavirus Disease (COVID-19) Dashboard. 2021. Available from: https://covid19.who.int/

NCBI Entrez Programming Utilities. Available from: https://www.ncbi.nlm.nih.gov/books/NBK179288/

Biopython Project. Available from: https://biopython.org/

NumPy. Available from: https://numpy.org/

Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997; 9(8):1735-80. doi: 10.1162/neco.1997.9.8.1735

Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980. 2014.

Chollet F. Deep Learning with Python. Manning Publications Co. 2018.