Neural Network Parameter Estimation for COVID-19 with Vaccinated and Timely-Delayed Diagnosis

Samuel Kwame Dassi
Department of Mathematics,
University of Cape Coast, Ghana

Friday, December 22, 2023

Presentation Outline

1. Introduction
2. Model Formulation
3. Deep Neural Network in Prediction
4. Results and Discussions
5. Conclusions and Challenges

Outline

Introduction

Brief Overview of the Project

- The goal of the project is to use machine learning and neural networks to study the behaviour of Covid 19 in Ghana
- Data was generated for 200 days with 1000 data point
- Python Programming language was used to do all the analysis and simulations
- Deep Neural Network was deployed to study the behaviour of the COVID-19 pandemic in Ghana
- Mean Squared Error (MSE) was used to evaluate the effectiveness of the model

Brief History

- Coronavirus Disease 2019 (COVID-19) is an infectious disease causing severe respiratory effects on patients.
- It was first identified in December 2019 in Wuhan, China.
- Common symptoms include fever, cough, fatigue, shortness of breath
- In March 2020, The WHO declared the virus as a pandemic.

In Ghana

- First two cases were recorded on 12th March 2020. (2 Ghanaians returning from Norway and Turkey)
- Measures taken to combat the virus include widespread testing, contact tracing, treatment protocols, and public health campaigns to promote preventive measures.
- As of June 2023, Ghana recorded about 171,653 confirmed cases and 1,462 deaths due to COVID-19.

Outline

Model Formulation

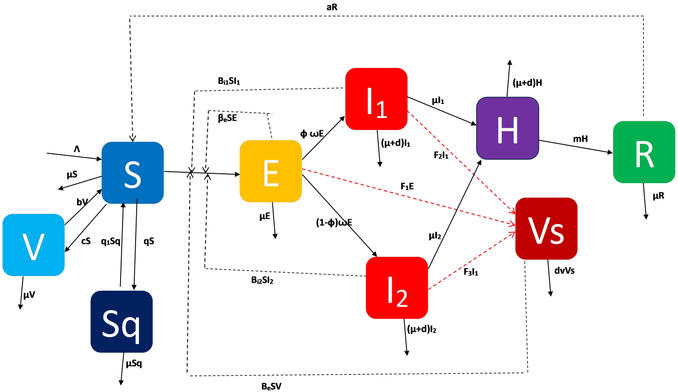


Figure 1: Flow Diagram for VSSqEI1I2HRVsS covid 19 model

This gives a system of Ordinary Differential Equations (ODEs):

$$\begin{aligned}
\frac{dV}{dt} &= aS - \mu V - bV \\
\frac{dS}{dt} &= \Lambda + bV - \lambda S + q1Sq - (\mu + q)S + cR \\
\frac{dSq}{dt} &= qS - q1Sq - \mu Sq \\
\frac{dE}{dt} &= \lambda S - \omega E - \mu E \\
\frac{dI1}{dt} &= \phi\omega E - \gamma 1I1 - \mu I1 - dI1 \\
\frac{dI2}{dt} &= (1-\phi)\omega E - \gamma 2I2 - \mu I2 - dI2 \\
\frac{dH}{dt} &= \gamma 1I1 + \gamma 2I2 - mH - \mu H - dH \\
\frac{dR}{dt} &= mH - \mu R - cR
\end{aligned} \tag{1}$$

$$\frac{dVs}{dt} = f1E + f2I1 + f3I2 - dvVs$$

(2)

Table 1: Compartmental Explanations

Compartments	Explanations
V	Vaccinated class
S	Susceptible
Sq	Self-quarantine susceptible
E	Exposed
I1	Infectious with timely diagnosis
I2	Infectious with delayed diagnosis
H	Hospitalized
Vs	The viral spread in the environment
R	Recovered
N	Total human population

Parameter Setting for the model

The parameters used in the model are indicated in the table below:

Table 2: Parameters and Values from literature

Parameters	Values	Parameters	Values
a	0.73589	βe	6.0380e-8
b	0.32	$\beta i1$	3.8196e-8
c	0.936	$\beta i2$	1.4286e-5
Λ	1319.294	Φ	0.9000
q	0.0333	$\gamma 1$	0.5000
q1	1.6945e-5	$\gamma 2$	0.0714
d	0.006139	m	4.2578e-5
f1	0.0178	βv	4.00199e-8
f2	0.3115	ω	$1/5.2$
f3	4.6131e-5	μ	4.2578e-5
dv	0.3117		

Data

- Data is a key ingredient of every model
- The data used in the various model consists of the COVID-19 data generated over a period of 200 periods (days)
- From day 0 to day 200, 1000 data points were generated using Python
- Everyone started out as susceptible except for those initially exposed, infected, and hospitalized.

Outline

Neural Network in predicting Covid 19 Spread

What is Neural Network

- Neural networks are a type of machine learning algorithm that are modeled after the way the human brain works.
- They consist of layers of interconnected nodes, or artificial neurons, that process and transmit information.
- The first layer receives input data, such as an image or text, and each subsequent layer processes the output from the previous layer until the final layer produces an output, which could be a prediction of a class or a numerical value.
- Neural networks are used to recognize patterns, process data, and make predictions.

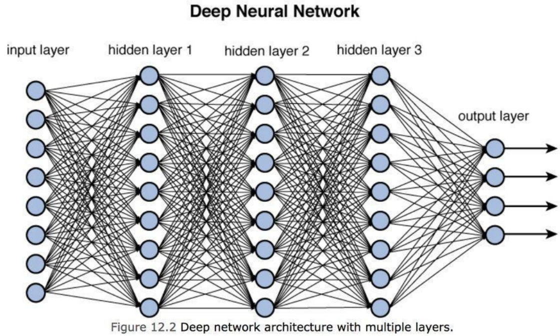


Figure 2: The neural network with multiple layers

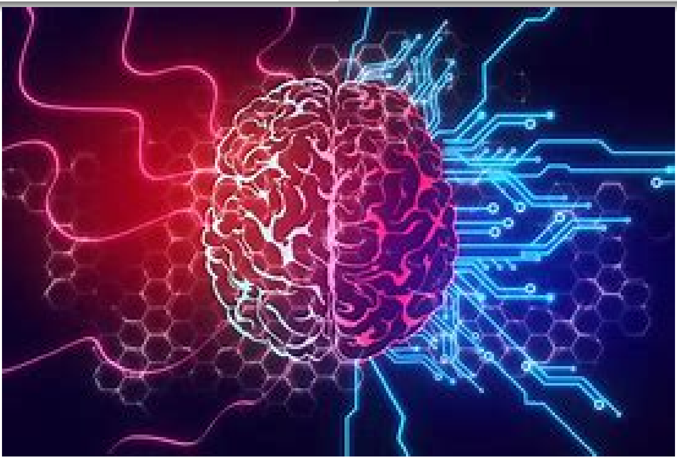


Figure 3: The Neural network works as the brain functions

How the Network Works

- During the training process, the weights and biases of the connections between the nodes are adjusted so that the network can learn to accurately predict the output for the input data.
- This is done by comparing the predicted output to the actual output and using an optimization algorithm to minimize the difference between the predicted and actual output.
- Through this iterative process of adjusting the weights and biases, the neural network learns to recognize patterns and make predictions based on the input data it receives.
- Once the network is trained, it can be used to make predictions on new, unseen data

Components of Neural Network

- Input: The data that is fed into the network for processing
- weights: The set of parameters that determine the strength of connections between neurons in the network.
- activation function: This is a crucial component of each artificial neuron. It introduces non-linearity to the network, allowing it to learn and approximate complex relationships in the data
- output: This typically represents the predicted values or classes for the given inputs

The Activation Function Used in the Model

- Rectified Linear Unit (ReLU): ReLU sets negative values to zero and keeps positive values unchanged. It is widely used due to its simplicity and ability to address the vanishing gradient problem.
  $\text{ReLU}(\text{x}) = \max(0, x)$

- Hyperbolic Tangent (tanh): Tanh squashes the output between -1 and 1, making it useful for classification problems that need outputs ranging from negative to positive.

$$\text{f}(\text{x}) = \frac{2}{1 + e^{-2x}} - 1$$

Forward Propagation During forward propagation, the input data is fed into the neural network, and the activation of each of the eight hidden layers are calculated successively until the final output is obtained. The output of forward propagation is the prediction made by the neural network for the given input data

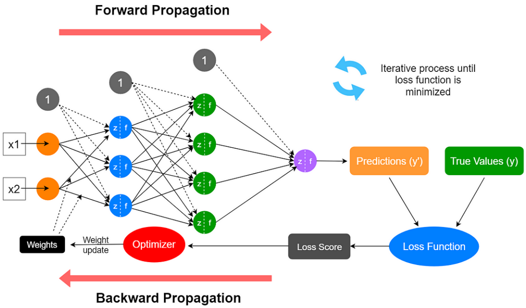


Figure 4: The Forward Propagation Process of Neural Networks

Backward Propagation (Backpropagation) After forward propagation, the neural network's prediction is compared to the actual target values to compute the error or loss. Backpropagation is used to update the network's parameters (weights and biases) in a direction that minimizes the error.

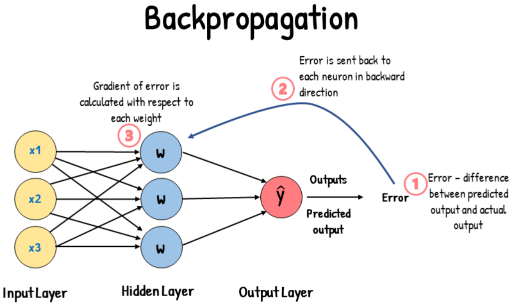


Figure 5: The Backward Propagation Process of Neural Networks

Learning Rate used in Training the Neural Network Model

- The learning rate is a hyperparameter used in training machine learning models, particularly in gradient-based optimization algorithms, to control the step size at each iteration during the optimization process
- the learning rate decides how quickly or slowly a model learns from the data during training
- Common learning rate values are between 0.000001 and 0.1
- For this project, a learning rate of 0.001 was defined

Scheduling the Leaning Rate

- The CyclicLR scheduler was used to implement a cyclical learning rate policy that gradually varies the learning rate between the base learning rate of 0.000001 and the max learning rate of 0.1 over a cycle.
- The cycle is determined by the step size of 4000, which indicates the number of steps to increase the learning rate from the base learning rate to the max learning rate.

Optimizer used to determine the Leaning Rate

- An optimizer is an algorithm used to optimize or update the parameters of a model during the training process. The goal of optimization is to find the set of model parameters that minimize the loss function.
- The Adam optimizer is commonly used to optimize the parameters of neural networks during the training process. It is an adaptive learning rate optimization algorithm that combines the benefits of different algorithms
- This optimizer maintains a separate learning rate for each parameter and uses estimates of the mean and variance of the gradients.

Loss Function

- A loss function is used to measure the difference between predicted and actual output.
- A common loss functions; mean squared error (MSE) was used to evaluate the model.
- The primary goal during the training process is to minimize the value of the loss.
- Gradient descent as an optimization algorithm used to minimize the loss function.
- By minimizing the loss, the neural network learns to make better predictions.

Outline

Numerical Results

- The COVID-19 model is again illustrated.

$$
\begin{aligned}
\frac{dV}{dt} &= aS - \mu V - bV \\
\frac{dS}{dt} &= \Lambda + bV - \lambda S + q_1 Sq - (\mu + q)S + cR \\
\frac{dS_q}{dt} &= qS - q_1 Sq - \mu Sq \\
\frac{dE}{dt} &= \lambda S - \omega E - \mu E \\
\frac{dI_1}{dt} &= \phi\omega E - \gamma_1 I_1 - \mu I_1 - dI_1 \\
\frac{dI_2}{dt} &= (1-\phi)\omega E - \gamma_2 I_2 - \mu I_2 - dI_2 \\
\frac{dH}{dt} &= \gamma_1 I_1 + \gamma_2 I_2 - mH - \mu H - dH
\end{aligned} \tag{3}
$$

- The graph of the induced compartments of the ODEs is presented.
- The results from the analysis show that most of the disease-induced compartments peaked as early as within the first ten days.
- The number of the exposed population was falling but still displayed a high level.
- While the vaccination class drastically fell to almost 0 after a few days, the infectious and hospitalized population still need to be handled well in order to avoid another surge.

Table 3: Literature Parameters and Values

Parameters	Values	Parameters	Values
a	0.73589	β_e	6.0380e-8
b	0.32	β_{i1}	3.8196e-8
c	0.936	β_{i2}	1.4286e-5
Λ	1319.294	Φ	0.9000
q	0.0333	γ_1	0.5000
q_1	1.6945e-5	γ_2	0.0714
d	0.006139	m	4.2578e-5
f_1	0.0178	β_v	4.00199e-8
f_2	0.3115	ω	1/5.2
f_3	4.6131e-5	μ	4.2578e-5
d_v	0.3117		

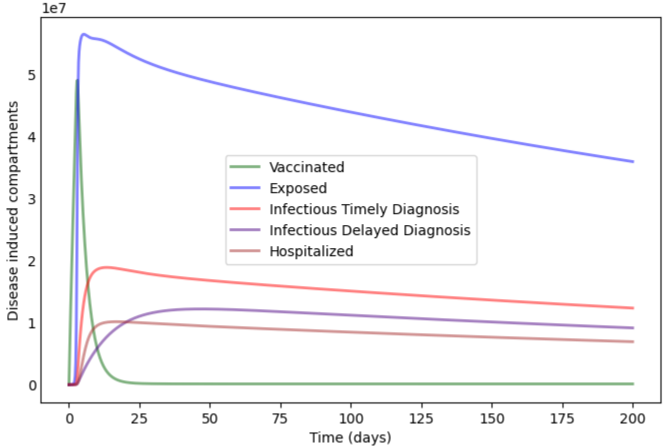


Figure 6: Disease induced compartments

Parameter Estimation From Neural Network

- To obtain the optimum parameter values for the network, learnable parameters were set up using Python's library called PyTorch.
- These parameters were optimized during training to find the best values that minimized the error and improved the model's performance.
- At every iteration during training, the estimated parameters were compared to the initial parameter values from literature.
- The new parameters estimated from the neural network training is summarized in Table 4.3 as compared with their corresponding literature parameters.

Table 4: Comparison of Literature parameters and Neural Network (NN) Parameters

Parameters	Literature Values	NN Values
a	0.73589	0.56949
b	0.32	0.3316
c	0.936	0.6511
Λ	1319.294	0.2726
q	0.0333	0.6133
q_1	1.6945e-5	0.7294
d	0.006139	0.1841
f_1	0.0178	0.2841
f_2	0.3115	0.5517
f_3	4.6131e-5	0.6142
d_v	0.3117	0.6086
β_e	6.0380e-8	0.1044
β_{i1}	3.8196e-8	0.4950
β_{i2}	1.4286e-5	0.11405
Φ	0.9000	0.1941
[illegible]	0.5000	0.2238

- The result from these new NN parameter sets is presented for the disease-induced compartments for 300 days.

- While the Exposed classed plateaued early, those in the timely diagnosis class were not as much as those in the delayed class.

- The Hospitalized class was the last to plateau. This may be due to the insufficient hospital facilities in the country.

- Also the drug for treatment of Covid-19 patients was unavailable in the initial periods of the pandemic.

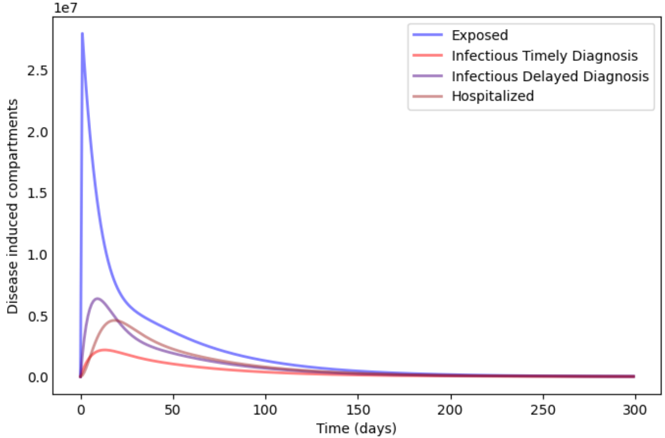
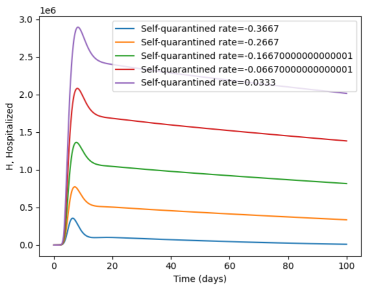


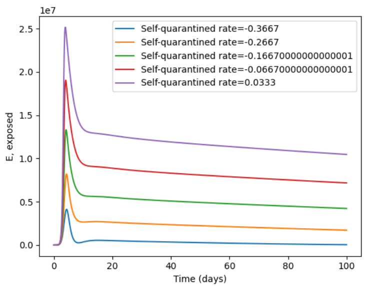
Figure 7: Disease induced compartments for the neural network parameter estimation

Sensitivity Analysis

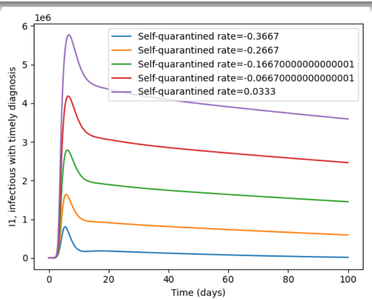
Effects of some NN parameters on the disease-induced compartments were analyzed and presented. The parameters include the rate of self-quarantine (q), vaccination rate (a), and recruitment rate (Λ).



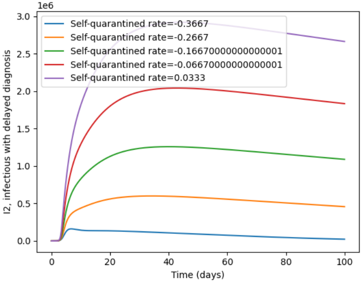
Hospitalized



Exposed

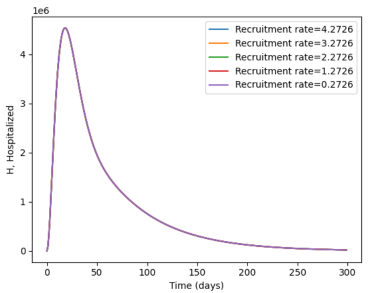


Timely Diagnosis

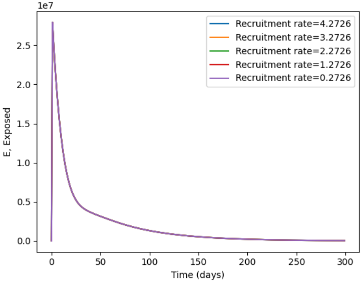


Delayed Diagnosis

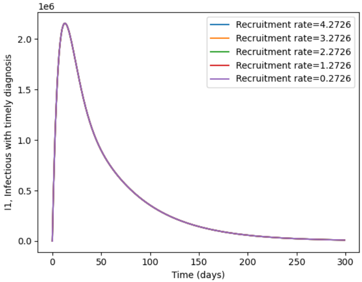
Figure 8: Effect of changes in the self-quarantined rate on Hospitalized, Exposed, Timely and Delayed Diagnosis classes



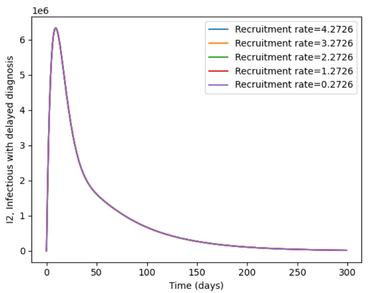
Hospitalized



Exposed

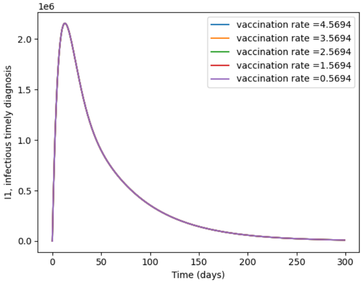


Timely Diagnosis

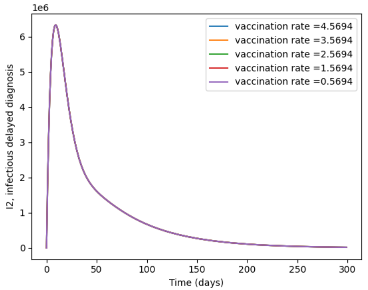


Delayed Diagnosis

Figure 9: Effect of changes in the recruitment rate on Hospitalized, Exposed, Timely and Delayed Diagnosis classes



Timely Diagnosis



Delayed Diagnosis

Figure 10: Effect of changes in the Vaccination rate on Timely and Delayed Diagnosis classes

Covid 19 Prediction Using Neural Network Parameters

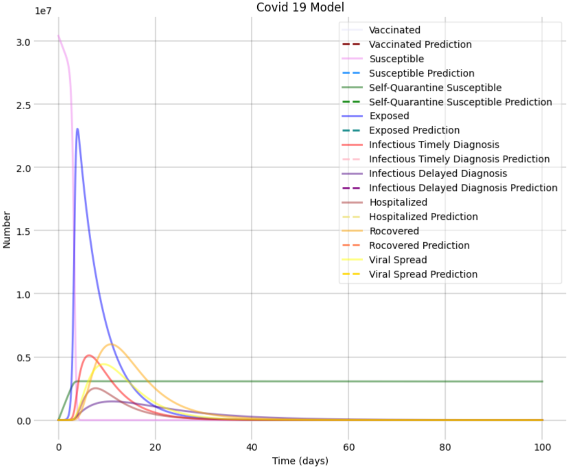


Figure 11: Predictions of Covid-19: Predicted and Actual Output

Model Evaluation

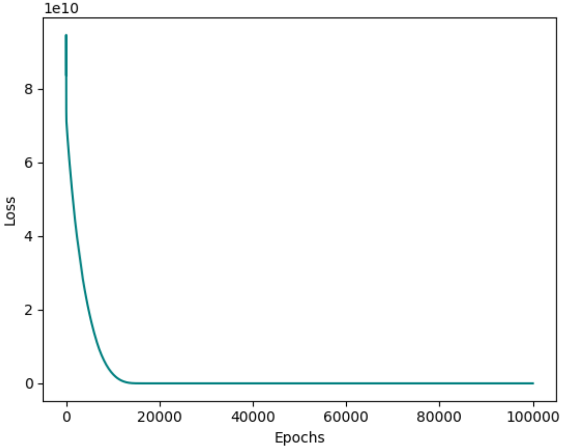


Figure 12: Loss curve vs epochs

Outline

Conclusions

1. The focus of this project was to develop a neural network parameter estimation model for COVID-19 with vaccinated and timely-delayed diagnosis.
2. I aimed to develop a model that could accurately estimate the parameters of the COVID-19 transmission dynamics considering the effects of vaccination and timely diagnosis.
3. The results obtained highlight the importance of considering vaccination and timely diagnosis in modeling COVID-19. since both have significant impacts on the transmission dynamics of the disease.
4. The insights gained from this project can contribute to the development of effective strategies for disease control and prevention.

Challenges

1. The model does not produce much sensitivity to the parameters estimated from the neural network
2. Nonetheless, the project demonstrates the potential of neural network parameter estimation techniques in understanding and modeling the transmission dynamics of COVID-19, considering the influence of vaccination and timely diagnosis.
3. One key challenge faced during this project was the availability of computing resources.

THANK YOU

Samuel Kwame Dassi
sdassi002@stu.ucc.edu.gh
profskdassi@gmail.com
+233240028612