Recurrent Neural Network in Glucose Prediction

Anna Zhang

October 2020

Abstract

This paper aims to build a model for glucose prediction using the preferred architecture. According to National Diabetes Statistics Report from Centers for Disease Control (CDC), by 2020, there are 34.2 million Americans with Diabetes, which is 1 in every 10 people. Therefore, glucose prediction, with its potential use in "artificial pancreas", has become crucial in the realm of managing patients' glucose level and taking necessary precautions. In this paper, we develop a glucose prediction algorithm for patients to predict their blood sugar level in the future 30 minutes and 60 minutes with the input data of last 4 hours. The general data set from a Continuous Glucose Monitoring (CGM) simulator is of three input fields – glucose levels, insulin, and time. The model is built by Recurrent Neural Network (RNN) and Long Short-Term Memory network (LSTM). Eventually, the results are evaluated by the mean squared error (MSE) between the actual blood sugar levels and the predicted ones. The mean value of the best MSE (unscaled) of the three out of ten patients in 30 minutes is 0.0138, while the mean value in 60 minutes is 0.0093. From the results, we can see that the performance of the prediction model is fairly accurate. As a result, we can conclude that applying LSTM to the task of predicting blood sugar level in the future is a favorable choice.

1 Introduction

In this paper, we use the model built by Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) to achieve the goal to predict glucose level. This approach requires patients' data of the past 4 hours in order to predict their blood sugar level in the next 30 and 60 minutes.

Machine Learning (ML) is the science of getting computers to act without being explicitly programmed [1]. It learns and extracts patterns from data, and the model discovers mappings from the presentation of input data to the output [2]. However, this traditional machine learning algorithm has its drawbacks, which requires domain expertise and needs problem statements to break down to different parts to be solved first before combining the results at the final stage [3]. A deep learning algorithm, on the other hand, is created by Deep Neural Networks (DNN). DNN contributes greatly to glucose prediction because the deep learning techniques solve problems end to end, eliminating the difficulty to break down to different sections [3]. For another thing, it tries to learn high-level features from data in an incremental manner [3], in order to maximize the performance with a large amount of data. Here is a graph comparing the performance of traditional machine learning algorithms and DNN [3]:

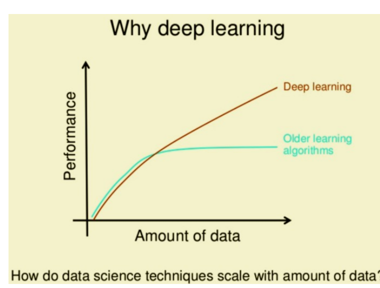

Figure 1: Traditional Machine Learning vs. Deep Learning Algorithms

Researchers face some potential problems when it comes to forecasting the blood sugar level. Data in the real world includes noise, as some subjects might encounter difficulty collecting their data, and individuals have insulin variability [2]. Therefore, there might some inconsistency in those data. Since glucose prediction is a time-series problem, RNN offers the necessary method, which is designed to recognize a data's sequential characteristics and use patterns to predict the next likely scenario [4].

In this paper, we build our model by RNN and LSTM. We then evaluate it using MSE. Our goal is to determine the model's performance in predicting glucose level in the future, so we collect 360-day data from the Continuous Glucose Monitoring (CGM) simulator instead of the real world to eliminate noise and outliers that the real-world data might have.

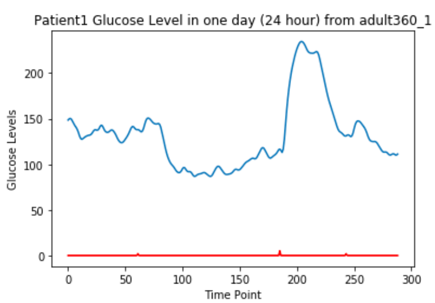

Figure 2: One Day Data from Patient 8

The graph shown above demonstrates the fluctuation of one Type 1 Diabetes (T1D) patient's glucose level in 24 hours. From the blue line in this graph, we can see that there are three time points when the glucose level increases by a large margin, indicating the three times of eating carbohydrates. They are followed by the decreasing glucose levels, which occur due to the injection of insulin after meal, as suggested from three protuberances in the red line. During the other time in this day, the blood sugar level goes up and down in a relatively mild degree.

2 Methodology

The source of training and testing data comes from the UVA/Padova T1D simulator, the only emulator for glucose level simulation approved by the Food and Drug Administration (FDA). There is data collected from ten patients in 360 days. Each patient holds the data of time, glucose level, and insulin injection amount. The data is collected once every five-minutes, with 288 time points per day and a total number of 103680. In the collected data, the difference between the glucose level and the insulin amount is quite apparent, with the difference between three-digit input for glucose level (at time = 371, insulin= 116.36mg/dL) and unidigit input for insulin (at time = 371, insulin = 2.26mg/dL). Therefore, we need to normalize the data so that they can be measured at the same scale. In addition, each feature can be seen equally important in the model. In this case, we choose Min-max normalization and scale the input data into a range from 0 to 1. Min-max normalization is a common and straightforward means when it comes to normalize data, as the minimum value of the feature gets transformed into 0, while the maximum value turns into 1 [5]. It makes sure that all the features have the same scale [5]. The formula of this idea is shown below [5]:

$$\frac{value - min}{max - min} \tag{1}$$

We apply a sliding window to the input in order to offer the model with past-four-hour data, which is 48 time points based on the fact that the simulated data is sampled every 5 minutes. Then the output data will be presented by the glucose level in both the next 30 and 60 minutes, which are 6 and 12 time points respectively.

Predicting glucose level involves in time-series solution, which leads us to use RNN to build our model. RNN allows previous outputs to be used as inputs while including hidden states [6].

In this architecture, RNN remembers weights applied on inputs. Moreover, it's influenced by the hidden state that represents the context based on previous inputs and outputs [6]. In this case, sequence becomes important because the same input could re-

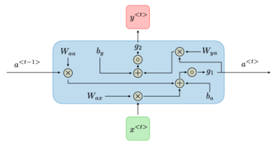

Figure 3: Architecture of Traditional RNNs

sult in a different output depending on the previous inputs and the hidden state. However, a traditional RNN has its drawbacks. Due to the constant model size regardless of the increasing size of input, it is difficult to access information from a long time ago as well as to predict future input from the current state [7]. As a result, we choose LSTM networks, which cope with the vanishing gradient challenge. As the graph below illustrates, the three gates help control the flow of information and solve the problem with long spans of prediction [8].

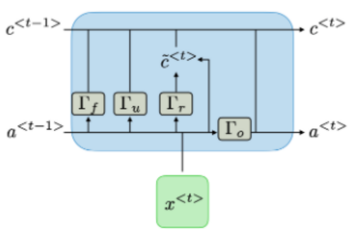

Figure 4: The Architecture of LSTM

In this work, we build the model with two layers of LSTM with a dropout value of 0.2 to avoid overfitting. A fully connected network is added at the final stage of the model. We try different numbers of layers of LSTM when optimizing our model, as we started with four layers. The deep network did show a slightly better performance. However, it also doubled the time to train the model. Based on the comparable performance of models with different numbers of layers, we decide to build our final model with two layers of LSTM. The difference in batch size also results in the various results in training the model. The smaller batch size (32) takes approximately three times as much time as the larger batch size (64) while demonstrating a less favorable outcome. Due to the consideration of time and effectiveness, we use 64 as the batch size for the model.

3 Results

The performance of how the model predicts glucose level is evaluated by Mean Squared Error (MSE), which is the most widely applied regression loss function to evaluate a model. MSE is the sum of squared distances between the true values of glucose level at certain time points and predicted values of the model. The formula of MSE is shown below:

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

Figure 5: Formula of Mean Square Error (MSE)

We run the model ten epochs for each patients, with a total number 100 epochs for ten patients. The results, demonstrated as the unscaled MSE, in the next 30 minutes are shown in the table and graph below.

30 min	
#	MSE
Patient 1	0.01686974836
Patient 2	0.01543427278
Patient 3	0.0205647427
Patient 4	0.01475629712
Patient 5	0.01771237616
Patient 6	0.0151163091
Patient 7	0.02801415896
Patient 8	0.01305677477
Patient 9	0.01370494038
Patient 10	0.01961117065

Figure 6: Prediction Results in 30 min evaluated by MSE

The three best results come from Patient 04, Patient 08, and Patient 09 (highlighted red in figure 6).

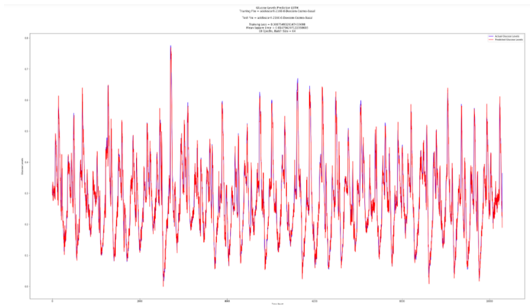

Figure 7: Glucose Prediction for 30 min Fitting for Patient 5

The MSE values are 0.0145, 0.0131, and 0.0137, respectively. The mean MSE value of the results for predicting glucose level in 30 minutes is 0.0138.

In order to demonstrate how well the predicted values fit with the actual data, the graph shown in Figure 7 is provided. The blue line indicates the actual value, while the red one suggests the predicted level. It's clear that the model carries out an effective performance due to the correspondence between the blue and red lines to predict the glucose level in 30 minutes.

Then we examine the results for prediction in 60 minutes. The results are presented in the table below – the best three performances come from Patient 02, Patient 08, and Patient 09 (highlighted red in Figure 8), with MSE values of 0.0109, 0.0089, and 0.0081, respectively. The mean of the results for predicting glucose level in 60 minutes is 0.0093.

Figure 9 is graphed based on the data from one of the finest results during the testing process. The graph shows that the prediction fits the values well. However, there might be some shortcomings in predicting the extrema, indicated by the higher points in blue line (maximum) at some time points and the lower points in the blue line (minimum) at the other points.

Based on the results of both tasks, it's apparent that Patient 08 and 09 perform better than the other subjects. There is one explanation to this trend – the fluctuation of the provided data does not include large data gap [2], and the pattern becomes regular for the model to track.

60 min	
#	MSE
Patient 1	0.01428677805
Patient 2	0.01086780852
Patient 3	0.01362253666
Patient 4	0.0162005043
Patient 5	0.01338504052
Patient 6	0.01108125648
Patient 7	0.02652136326
Patient 8	0.008996606355
Patient 9	0.008128890451
Patient 10	0.01737700218

Figure 8: Prediction Results in 60 min evaluated by MSE

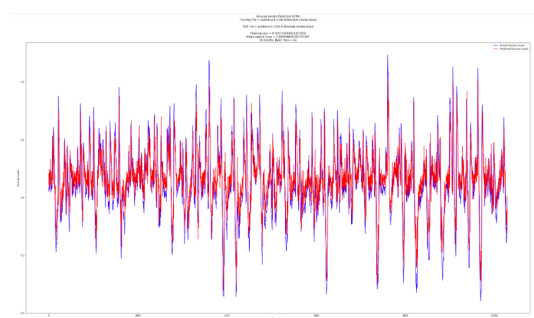

Figure 9: Glucose Prediction for 60 min Fitting for Patient 8

When we compare Figure 7 and Figure 9, we can see that the fitting for prediction in 30 minutes performs better and closer to the actual values of the subjects than the prediction in 60 minutes. However, based on the values of MSE, the performances of the model in forecasting glucose level in 30 and 60 minutes are capable of the given tasks. The fact that the performance becomes less favorable in a longer time span indicates the possibility of prediction delay [9] and the potential improvement of the model in the future.

4 Discussion and Conclusion

In this paper, a Recurrent Neural network (RNN) is designed as an effective method for glucose prediction. The model includes several layers of Long Short-Term Memory (LSTM) networks. The RNN is able to capture features from previous sequential data and offer the predicted glucose level in the upcoming 30 and 60 minutes. When comparing the results with other researches, although it's difficult to directly compare the mean squared error results due to the different parameters in the model such as the batch size and input size, we discover similar fitting patterns in the prediction. In this paper, we train the model with solely the glucose level and the amount of insulin injection. For future work, we can integrate other data fields such as the intake of carbohydrates to improve the model performance.

5 References

[1] A. Ng, "Machine Learning," Coursera, 2019.
[2] K. Li, J. Daniels, C. Liu, P. Herrero-Vinas and P. Georgiou, "Convolutional Recurrent Neural Networks for Glucose Prediction," in IEEE Journal of Biomedical and Health Informatics. doi: 10.1109/JBHI.2019.2908488

[3] S. Mahapatra, "Why Deep Learning over Traditional Machine Learning," Mar 2018.

[4] M. Rouse, "Recurrent Neural Networks," Jun 2018.

[5] C. Group, "Why Normalize," NYC, https://www.codecademy.com.

[6] A. Amidi and S. Admidi, "Architecture of a Traditional RNN," 2019, https://stanford.edu.

[7] A. Amidi and S. Admidi, "Handling Long Term Dependencies," 2019, https://stanford.edu.

[8] M. Lukosevicius and H. Jaeger, "Reservoir Computing Approaches to Recurrent Neural Network training," in ComputerScience Review 3.3, pp.127–149, May 2009.

[9] J. Reifman, S. Rajaraman, A. Gribok and WK. Ward, "Predictive Monitoring for Improved Management of Glucose Levels," J Diabetes Sci Technol, Jul 2007, doi: 10.1177/193229680700100405.