Predicting blood glucose level using RNNs

Diego Zavala
Stanford University
dzavalag@stanford.edu

Pedro Salgado
Stanford University
psalgado@stanford.edu

Abstract

Diabetes is one of the most restricting diseases of the XXI century, more than 100 million people in the US suffers from diabetes or prediabetes changing their life and behavior. The following report aims to contribute to current research by providing a deep dive into the applications of neural network in predicting the level of glucose in the blood in the next 30 and 60 minutes. By doing so, patients will be able to manage their insulin shots in a better way, improving their life standards.

After testing several methodologies, we concluded that using a RNN with a LTSM structure is the most efficient methodology to predict with a Root Mean Square Error (RMSE) of 24.6 mg/dL, which is significantly better than predicting through time-series or naive approaches.

1 Introduction

As defined by the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), diabetes is a "disease in which body's ability to produce or respond to hormone insulin is impaired, resulting in abnormal metabolism of carbohydrates and elevated levels of glucose in the blood and urine", which end up with health troubles limiting the life conditions of the patient.

Traditionally, diabetics use manual monitoring systems that include self-injections and manual blood glucose tests to manage the hormone's impairment, aiming to keep the blood glucose level within an acceptable range. However, in the past few years, there has been a significant introduction of technology substantially improving the living conditions of diabetic patients. Some examples are the CGMs (Continuous Glucose Monitoring system) and insulin pumps, that can infuse insulin into the body without requiring an injection.

Although technological developments, insulin pumps require manual processing by each patient when CGM device sends an alert if blood glucose level reaches a threshold. However, as the blood glucose level depends on the current level, on the recent insulin infusions and the carbs eaten, the goal of our project is to predict CGM in advance (30-45 minutes) and compare root mean square error (RMSE) of the different algorithms reviewed during the quarter.

The impact of the project is massive. According to Centers for Disease Control and Prevention (CDC), +100 million adults in the US are currently living with diabetes or prediabetes, trend that will keep growing due to the consumption behaviors of the population. We aim to provide solutions to ease people's life and improve the standard of living of millions.

2 Related work

Research in area has increase due to the application of deep learning methods. Venkatesan and Anitha (2006) introduced neural networks in the field, showing how supervised feed forward neural network

CS230: Deep Learning, Spring 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

with one hidden layer and Radial Basis Function (RBF) neural network outperformed the accuracy of traditional logistic regression models by 20%. Zecchin et al (2009) combined time series models with NN, reaching RMSE of 19.4 mg/dL on simulated data but obtained poor performances on out-of-sample observations. Raj (2018) founded that combining NN with Random Forests and Naïve Bayes classifiers can improve the accuracy on simulated data, but again out-of-sample performance was not as expected.

Lai et al (2018) proposed a comprehensive approach comparing above-mentioned methodologies against LSTNet, RNN and CNN in fields including traffic, solar energy, electricity and exchange rate. Based on the analysis of Lai et al (2018), Li, Daniels, Liu, Herrero and Georgiou (2018) proposed a convolutional RNN (CRNN) for Glucose Prediction, reaching 21.1 and 33.3 mg/dL RMSE for 30-minute and 60-minute out-of-sample predictions, respectively. Due to the impressive results, we use the later paper as benchmark for our analysis.

The main takeaways of the literature are:

- RMSE is the appropriate metric to compare accuracy, with results between 9 and 18 mg/dL for 30 minutes and 60 minutes, respectively.
- Data requires at least 3 months series and steps every 5 minutes to reach robust results.
- The total number of lags included in the model may vary, as so the prediction horizon, which fits between 30 minutes and 90 minutes.
- The state-of-the-art solution seem to be a combination of CNN and RNN.

3 Dataset and Features

As the input of our algorithm, we used Diego's personal CGM and insulin pump records from the last 4 months, which includes around 35,000 observations including:

- Blood glucose levels (every 5 minutes adding up to approx. 35k values)
- Insulin infusions, including the basal rate (not frequently adjusted) and "bolus" that varies according to the glucose level and/or number of carbs eaten
- Carbs eaten

The required data was collected and prepared to be used. The time period that will be used is from December 1st, 2018 to March 30, 2019. The process of data collection and preparation include the following steps:

- The first step is to upload the insulin pump information to the t:connect diabetes management application. This application is provided by Tandem, the pump manufacturer
- From the t:connect platform is possible to download a csv file with the raw data. Specifically, we downloaded 3 data tables: one with the glucose level observations (every 5 minutes), other with the basal rates (every time it changes), and other with the insulin bolus and carbs (every time a bolus was applied)
- The 3 tables were consolidated in 1 table, with observations every 5 minutes. Glucose level was interpolated when there was 1 one missing value, and left blank otherwise. The basal rates (that is in units of insulin per hour) was re calculated as a 5-minute rate, and the insulin bolus and carbs were allocated to the corresponding 5-minute slot

The output looks like this:

Before using it in the model, the data is scaled, randomized and splitted between train (72.25%), dev (12.75%) and test (15%) data

4 Methods

4.1 Loss Function

We are interested in predicting the glucose value as close as possible to the actual value. Given that, we are going to use mean squared error as the loss function.

	Time_EventDateTime	Id	Real	Date	CGM	Insulin_Basal	Insulin_bolus	Carbs	Unnamed: 9
0	2018-12-01 00:00:12	1	1	2018-12-01	118.0	0.10	0.0	0	NaN
1	2018-12-01 00:05:12	2	1	2018-12-01	127.0	0.09	0.0	0	NaN
2	2018-12-01 00:10:12	3	1	2018-12-01	130.0	0.09	0.0	0	NaN
3	2018-12-01 00:15:12	4	1	2018-12-01	128.0	0.09	0.0	0	NaN
4	2018-12-01 00:20:12	5	1	2018-12-01	124.0	0.09	0.0	0	NaN

Figure 1: Input Data

4.2 Optimizer and weight initialization

- Optimizer: Adam with beta_1=0.9 and beta_2=0.999
- Weight initialization: Xavier normal initializer

4.3 Baseline models

We use 2 different algorithms as baseline: Naive and ARIMA.

- The Naive algorithm is one of the most simple algorithm that can be used in a forecasting problem with time-series data. The prediction is the last value observed.
- ARIMA, or AutoRegressive Integrated Moving Average Model is a statistical model used in time-series forecasting models. It has 3 main parts: an auto-regression, integration by subtracting an observation from the previous one, and moving-averages.

4.4 Deep learning model: Recurrent neural network

Recurrent neural networks have been very succesful in problems with sequential inputs, such as time-series, text or audio. The most promising results have been accomplished using Long Short-Term Memory layers (LSTM), described originally by Sepp Hochreiter and Jurgen Schmidhuber. LSTM layers are able to learn long-term dependencies in the sequence.

The key element of a LSTM layer is the LSTM cell, that is represented by the following figure:

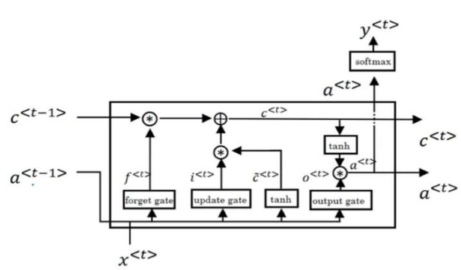

Figure 2: LSTM cell

Each cell has 3 gates: forget, update and output. The forget gate decides which information of the current state of the cell ($c^{<t-1>}$) is retained and which is forgotten, by taking into consideration $a^{<t-1>}$ and $x^{<t>}$. The update gate then decides the new information that will be stored in the cell state $c^{<t>}$. To do that, the gate outputs a number between 0 and 1 for each candidate value, that were created by a tanh layer. Finally, the output gate decides what will be the output of the cell.

The formulas for each gate are presented below:

$$
\begin{aligned}
\tilde{c}^{<t>} &= \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \\
\Gamma_u &= \sigma(W_u[\,a^{<t-1>}, x^{<t>}] + b_u) \\
\Gamma_f &= \sigma(W_f[\,a^{<t-1>}, x^{<t>}] + b_f) \\
\Gamma_o &= \sigma(W_o[\,a^{<t-1>}, x^{<t>}] + b_o) \\
c^{<t>} &= \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>} \\
a^{<t>} &= \Gamma_o * \tanh c^{<t>}
\end{aligned}
$$

The architecture of the network used is the following:

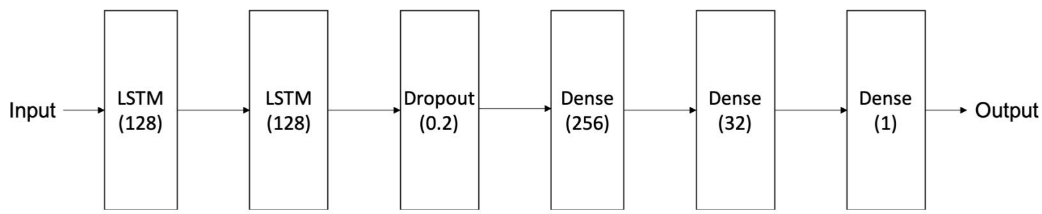

Figure 3: Architecture of the RNN used

This design was the best performing of the ones tested, that included combinations of Convolutional, LSTM, Bi-directional LSTM and fully conected layers.

5 Experiments, results and discussion

5.1 Hyperparameter tuning

Using the Keras framework, we searched for hyperparameters that improved the results of our model. The hyperparameters selected were:

- Learning rate: 0.001 (reduced when the metric has reached a plateau)
- Dropout: 0.2
- Mini-batch size: 4096 (input is only numeric, so it's possible to use a large size)

5.2 Results

Table 1: Results by Model

Method	Result (RMSE, mg/dL)
Naive	32.1
ARIMA	54.2
RNN	24.6

The model that showed the best results was the recurrent neural network, with a RMSE of 24.6 m/dL. The results obtained from our experiments were in line with our expectations, given that the data follows a stationary process, which was proven through a Dickey-Fuller test with a p-value of 1%. This implies that a model that takes lag observations as input should perform better in this task, which also is in line with the Naive model having a relatively good result.

Below we present a comparison of the RNN predictions with respect to the actual values for a subset of the data:

Compared to the related work, our model was not able to achieve the same level of accuracy as Li, Daniels, Liu, Herrero and Georgiou (2018) (RMSE of 21.1 mg/dL), but the results were pretty close (21.1 vs 24.6). As a reference, the mean of the data is close to 170 mg/dL, which means that the

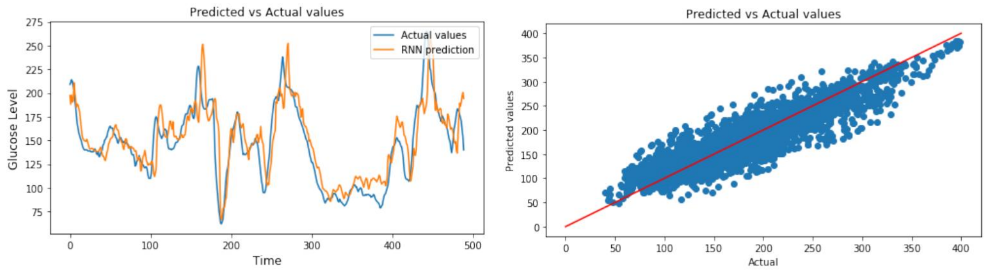

Figure 4: Predicted versus actual values in time

RMSE of our model was a 14.5% whereas the benchmark model was 12.4%. Regarding standard deviation, for the data is 60 mg/dL, which is more than 2 times the RMSE of the RNN.

Compared to the baseline models, the RNN is more accurate: the Naive model has a RMSE ⅓ larger than the RNN, and the ARIMA model more than 2 times larger.

6 Conclusion and Future Work

RNNs have shown promising results in problems related to forecasts using time-series data. In this project, we used a RNN to predict the blood glucose level of a diabetic person taking as input the previous blood glucose level, insulin infused and carbs eaten. The results of the RNN model are better than the baseline models (Naive and ARIMA), but not superior to those found in the literature.

Future work in this field might consider the use of several patients' data to train a general model than can be customized as a second step to a particular patient. This might improve the performance of the model. Another interesting field of study is the incorporation of physical activity indicators as inputs in the model. Currently, the models reviewed don't take into account the effect of exercise and other physical activity in the blood glucose level. Using information from wearable devices, such as a Fitbit or Apple Watch, could be a way to incorporate this effect.

7 Contributions

Diego: Data sourcing, data cleaning and analysis, and implementation of the algorithms

Pedro: Data architecture, time series analysis, bibliography research

References

[1] Anitha, P. V. (2006). Application of a radial basis function neural. CURRENT SCIENCE, VOL. 91, NO. 9.

[2] C. Zecchin, A. F. (2009). A new neural network approach for short-term glucose prediction using continuous glucose monitoring time-series and meal information. 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society.

[3] Guokun Lai, W.-C. C. (2018). Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. 41st International ACM SIGIR Conference on Research Development in Information Retrieval.

[4] Hochreiter, S., Schmidhuber, J. (1997). . Long short-term memory. Neural computation, vol. 9, no 8, p. 1735-1780.

[5] Kezhi Li, J. D. (2018). Convolutional Recurrent Neural Networks for Glucose Prediction. [online] Available: https://arxiv.org/abs/1807.03043.

[6] Raj, M. (2018). Diabetes Prediction — Artificial Neural Network Experimentation.

GitHub

`https://github.com/diegozavala/RNN_diabetes/`