Recurrent Neural Network (RNN) Glucose Prediction Model

By: Anna Zhang

Purpose

This research aims to build a model with the most ideal neural network to predict future glucose level in the next 30 minutes and 60 minutes. The model can help diabetes patients to monitor their upcoming blood sugar level in case they need to take any precautions due to extreme glucose level.

Materials and Tools

UVA/Padova T1D Simulator

This is the only simulator approved by FDA and it is used to generate input data for the model.

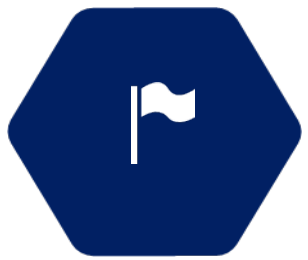
- data collected from 10 subjects in 360 days
- Collected once every 5 minutes, 288 time points per day, 103,680 time points in total
- Input field: time, glucose level, insulin injection amount
- Data imported to Python as “.mat” file

TensorFlow/Python

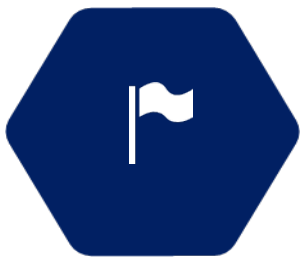
This is used as the main platform for processing data, programming and building the model.

- A Python software library
- For machine learning, particularly deep neural networks

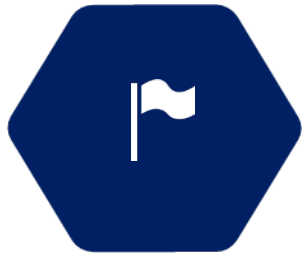
Plans and Procedures



Gather Data from the UVA/Padova simulator



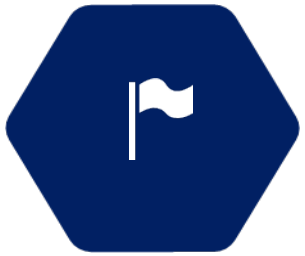
Normalize Data with Minimum-Maximum Normalization



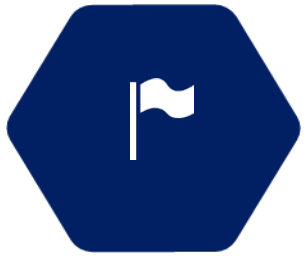
Split the normalized data into training set and testing set

Data Processing

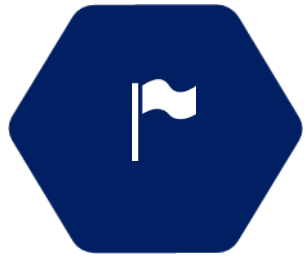
Plans and Procedures



Build the model



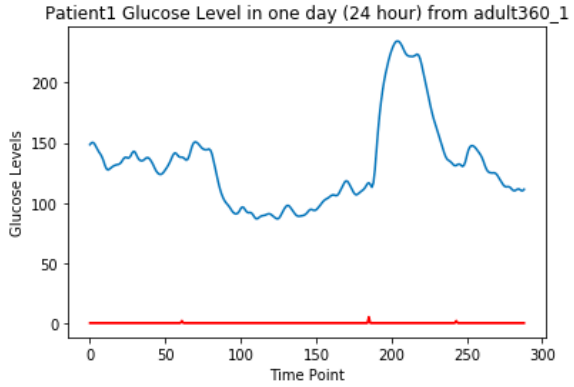
Train the model with training data (optimize parameters)



Test the model with testing data (use both MSE and graphs to evaluate the performance)

Preparation work

What does the data look like?

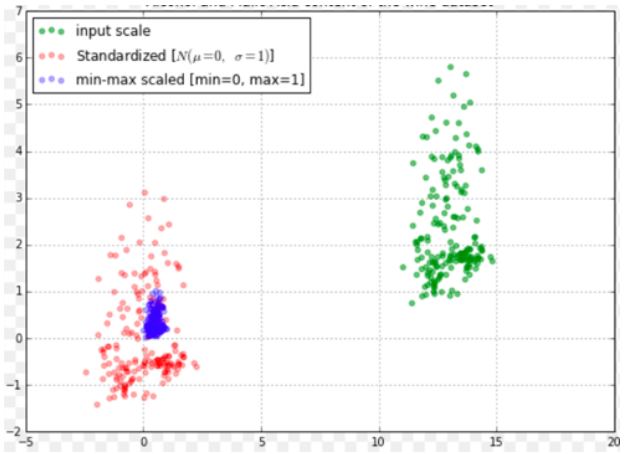


Data Collection

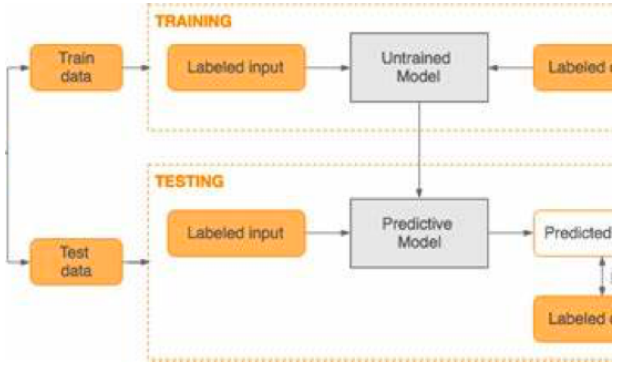
Data Normalization

Training&Testing datasets

Why do we need to normalize our data?



Why is splitting data into two datasets important in the research?



Major Timeline for My Project

Determine the neural network for my project

15 August, 2020

Collect Data

29 August, 2020

Import data into Python & Learn the pattern

30 August – 6 September, 2020

Write codes to normalize data

7 September - 13 September, 2020

Write codes to split training and testing data

13 September – 15 September, 2020

Build the model & Training the model (while optimizing parameters)

15 September – 29 September, 2020

Test the model & Record Data

1 October – 3 October, 2020

Improvement & Paper Work

3 October – 31 October, 2020

System Results

30 min vs. 60 min (LSTM)

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	
		Average: 0.01383933742

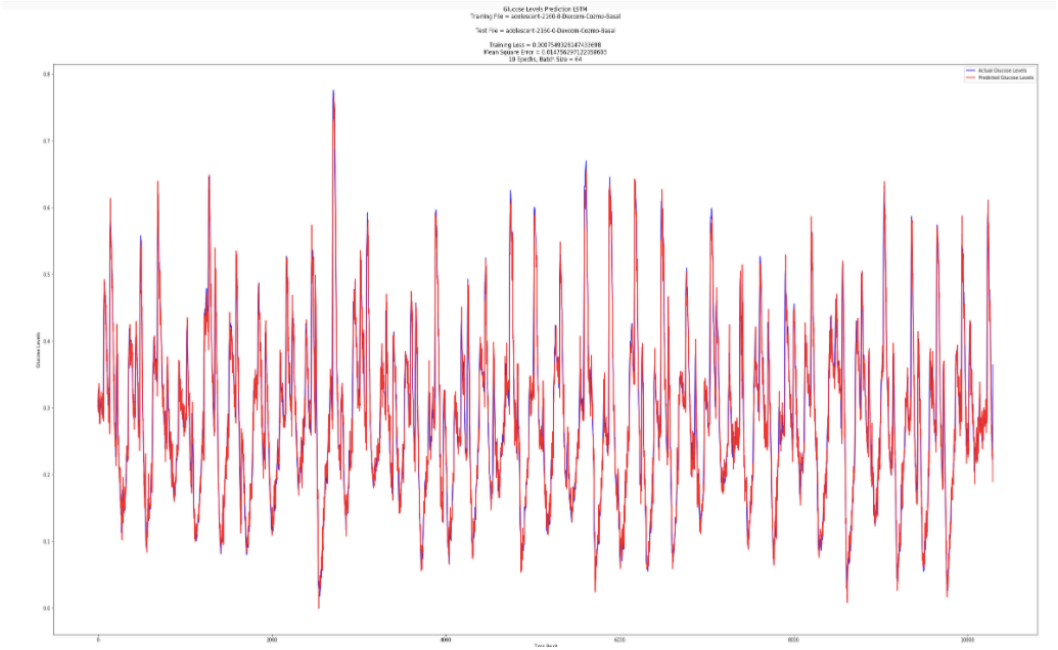
60 min		
#	MSE	
Patient 1	0.01428677805	
Patient 2	0.01086780852	
Patient 3	0.01362253666	
Patient 4	0.0162005043	
Patient 5	0.01338504052	
Patient 6	0.01108125648	
Patient 7	0.02652136326	
Patient 8	0.008996606355	
Patient 9	0.008128890451	
Patient 10	0.01737700218	
		Average: 0.009331101775

- The three smallest value of MSE is highlighted red.
- “Average” is calculated as the mean of the three smallest MSE values.

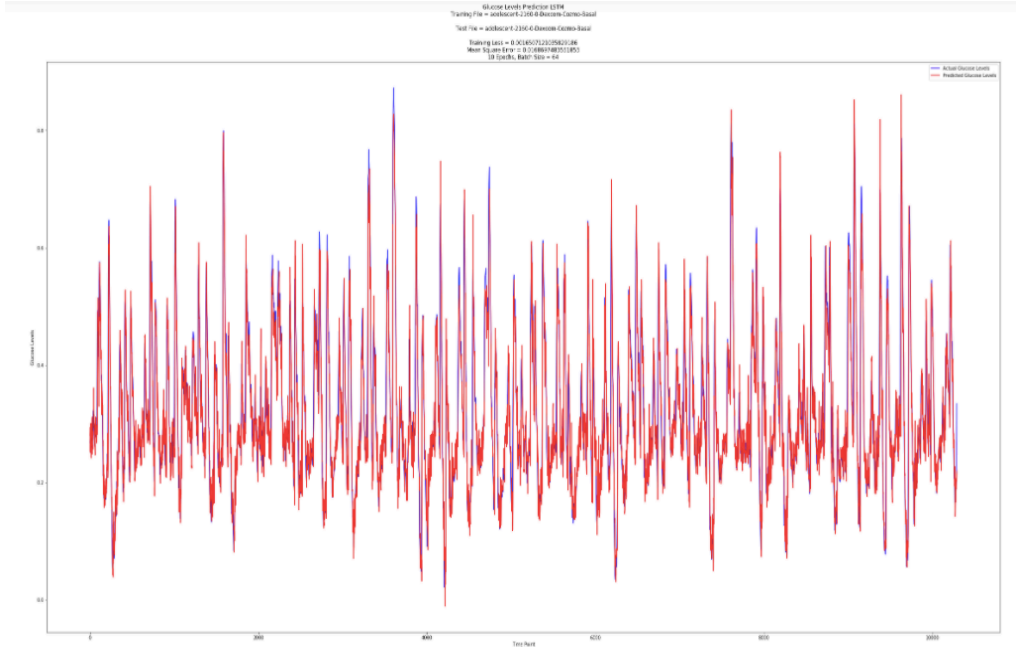
System Results

30 min vs. 60 min (LSTM)

- Red line is the predicted glucose level. Blue line is the actual value.



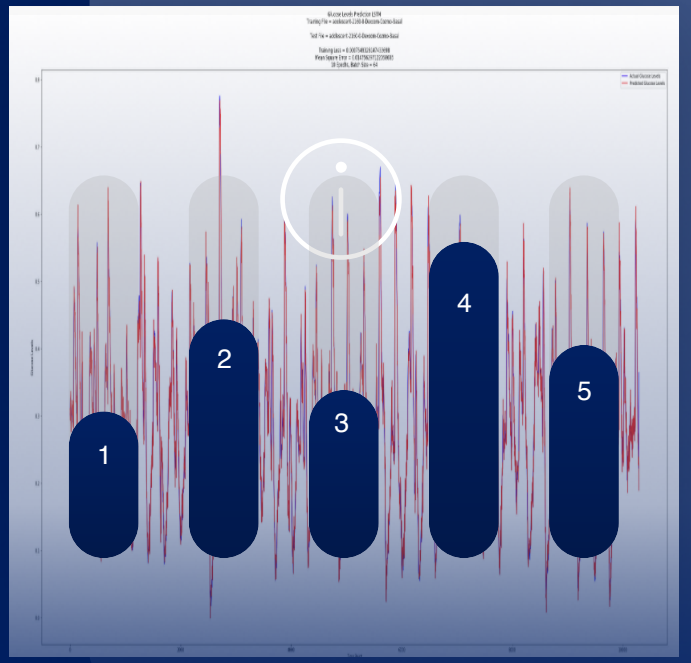
30 min



60 min

Analysis

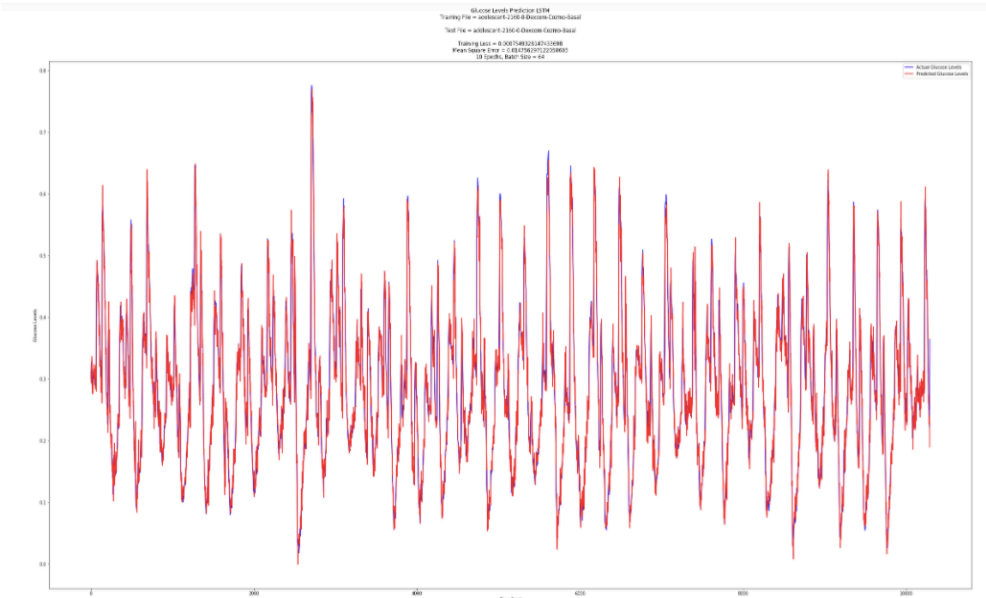
By comparing the average number of MSE in the three best performance in 30 min and 60 min, we see a fairly similar MSE value (0.0138 vs. 0.0093).



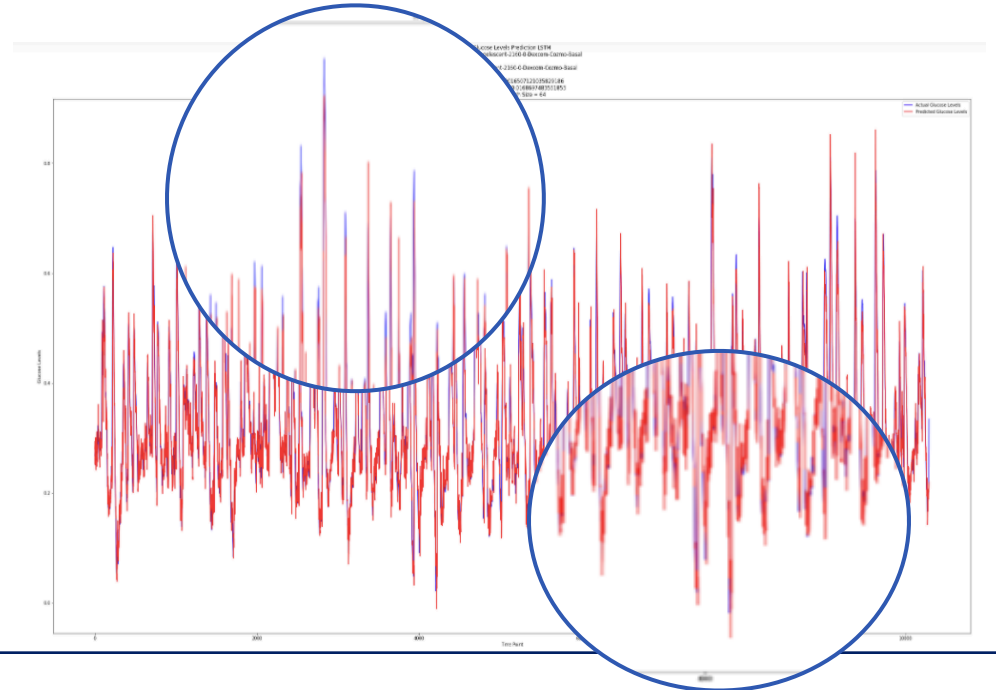
What can we see from the results?

Analysis

The red line and blue line in both graphs correspond to each other well. However, the 60-min graph does not deal with extreme values well, while the 30-min graph illustrates a more accurate prediction.



30 min

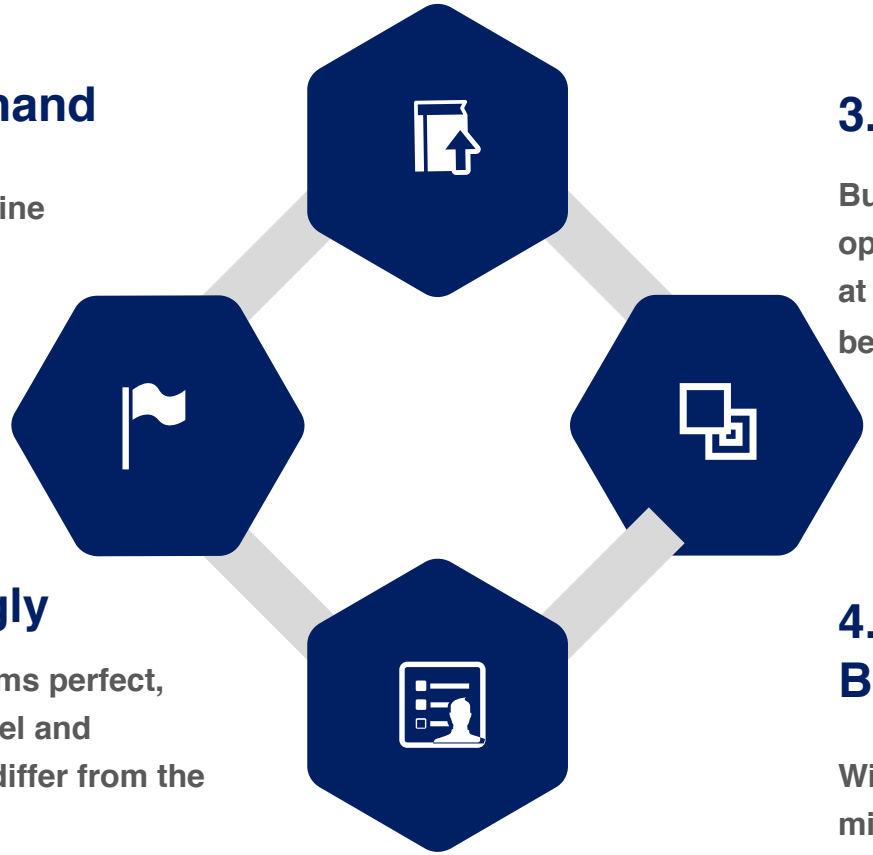


60 min

What have I learned from this project?

1. Prepare beforehand

Without the data, the machine cannot train themselves.



2. Plan Accordingly

Even though the plan seems perfect, time for building the model and debugging might always differ from the plan.

3. Always look for better ways

Building the model requires time to optimize parameters, as we never know at what value would our best parameter be.

4. Process needs to be enjoyable, But result is also important.

Without the result, the whole process might only costs time but gives nothing concrete.

Future Work

```
Epoch 1/10
1457/1457 [==============================] - 43s 29ms/step - loss: 0.0088
Epoch 2/10
1457/1457 [==============================] - 43s 30ms/step - loss: 0.0061
Epoch 3/10
1457/1457 [==============================] - 44s 30ms/step - loss: 0.0055
Epoch 4/10
1457/1457 [==============================] - 43s 30ms/step - loss: 0.0051
Epoch 5/10
1457/1457 [==============================] - 44s 30ms/step - loss: 0.0049
Epoch 6/10
1457/1457 [==============================] - 44s 30ms/step - loss: 0.0047
Epoch 7/10
1457/1457 [==============================] - 44s 30ms/step - loss: 0.0047
Epoch 8/10
1457/1457 [==============================] - 45s 31ms/step - loss: 0.0046
Epoch 9/10
1457/1457 [==============================] - 44s 30ms/step - loss: 0.0046
Epoch 10/10
1457/1457 [==============================] - 44s 30ms/step - loss: 0.0045

Unscaled Loss = 0.004390481393784285
```

- Use real-world data (with noises) to build a glucose prediction model.
- Build a hypoglycemia prediction model with data from UVA/Padova simulator.
- Apply real-world data to a hypoglycemia prediction model.

Thank you for listening!