01.116 Healthcare and AI Lab 2: Clinical Time Series Analysis

January 2021

1 Introduction

In this lab we would be studying blood glucose measure and forecasting for diabetics. Traditional blood glucose Monitoring for diabetes has always been tedious and cumbersome, where patients are often required to draw blood which is then fed into a blood glucose monitoring device, that outputs their glucose reading for that instance. This allows patients to get feedback on their glucose reading.

In recent advancement, continuous glucose monitoring (CGM) is becoming more common, where monitoring devices can be attached onto a patients (traditionally their arm), allowing them to actively monitor their blood glucose level.

Although CGMs have improve the process of glucose monitoring, it is still limited to a real-time feed. Here we attempt to introduce AI processes, in specific we introduce Recurrent Neural Network(RNN)/Long short term network(LSTM) to predict future output , allowing early detection of any potential disease complication.

A rough interpretation of blood glucose levels for diabetics according to https://www.medicinenet.com/

1. 70 milligrams per deciliter (mg/dL), you will usually have symptoms of low blood sugar.
2. after eating should be a range of about 100 mg/dL or less and 180 mg/dL about 2 hours after eating.
3. High blood sugar for those being treated for diabetes, the high range begins at 180 mg/dL
4. These people may have a fasting range of about 100 mg/dL or less and 180 mg/dL about 2 hours after eating.

2 CGM Dataset

In this dataset we will be provided with dataset containing the following columns of importance

1. RecID- unique record ID
2. DeviceDtTmDaysFromEroll (Date time index)
3. GlucoseValue (mg/dL) - the output glucose number

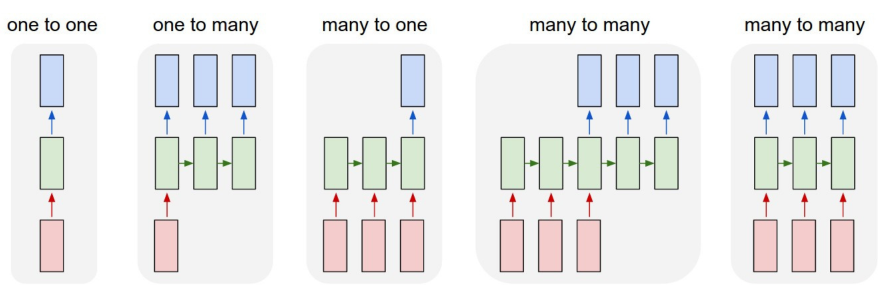
Prepare the dataset, filter out the necessary data and split it to its respective test and training sets.

Checkpoint1: Study the training data-set and answer the following 1) How many sets of ***continuous training data sets*** do you have available 2) The count of samples points per training data sets. Segments the data-sets into training and testing.

3 Data Pre-processing

We would be constructing a Many-to-one LSTM architecture , as such we require the data to be pre-processed into the necessary window size for training. Note that the training and validation data would expect the following structure:

1. Input data ("Many")
2. Expected data ("One")



Tip: you may want to specify a certain window size to structure your many-to-one split.

Checkpoint2: Display the dimensions for your pre-processed data and explain how it the window size is incorporated into this structure.
Checkpoint3: Explain how this many-to-one structure is presented is incomperated in your data structure.

4 Modelling & Training

We would now construct the LSTM model. Lucky for us , Tensorflow has a in-build module that allows us to easily construct this.

Useful Syntax: tf.keras.layers.LSTM(.)

Similarly to PS1, we would now require the following for calculating the loss and carry out the backpropogation

1. Optimiser
2. Loss Function

Checkpoint 4: Select the correct Loss Function and optimiser and explain the reason for your choice
Checkpoint 5: Graph and display the training loss of the model.

5 Validation

Now that we have our trained model , we would want to validate it on the rest of the remaining data to ensure that our model is not only accurate on a single session data but for mutiple other instance. Utilise the model on the remaining data available and show that the model accuracy is acceptable.

Checkpoint 6: Graph the remaining instances and plot them (Provide the true data and the validated data on the same graph), display the mean loss for each of the instance.

6 Optimisation

We would now want to optimise the prediction accuracy by carrying out hyper-parameter optimisation. Vary the windows size of the model and find the optimal window size that gives back the lowest loss.

Checkpoint 7: Write a short report presenting your analysis on the optimal hyper-parameter of choice. You may include the necessary graphs or print out to explain your optimal hyper-parameter of choice.