

Predicción de los Niveles de Glucosa en Sangre para Pacientes Diabéticos a Partir de Redes
Neuronales Recurrentes Tipo LSTM

Ismael Vega Rojas

Trabajo de Grado para Optar al Título de Ingeniero Electrónico

Director

Rodolfo Villamizar Mejía

PhD. Tecnologías de la información

Codirector

Edward Alfonso Rodríguez Moreno

Msc. En Ingeniería Electrónica

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Ingeniería Electrónica

Bucaramanga

2023

Dedicatoria

A mi mamá Leonor Rojas Capacho, este logro es más de ella que mío. Fueron sus besos y abrazos llenos de sinceridad, su apoyo siempre incondicional, su sonrisa llenando mi corazón de felicidad, sus te amo mi mayor motivación, y su amor las fuerzas para seguir adelante.

Agradecimientos

A Dios, quien es la fuente de la sabiduría y poder, a mi mamá por su apoyo y comprensión permanente, a la UIS por tener el programa de Ingeniería Electrónica, al director del proyecto, el Doctor Rodolfo Villamizar Mejía y los demás profesores por su gran aporte, esmero y metodología al transmitir sus conocimientos, al codirector del proyecto Edward Alfonso Rodríguez Moreno, a mis compañeros de estudio por el trabajo en equipo y a todas aquellas personas que me ayudaron para lograr este objetivo.

Tabla de Contenido

	Pág.
Introducción	12
1. Marco conceptual	15
1.1 Diabetes	15
1.1.1 Diabetes mellitus tipo 1 (T1DM)	15
1.1.1.1 Modelo Uva Padova	16
1.2 Redes Neuronales Recurrentes tipo LSTM	17
2. Conjunto de datos	21
2.1 Obtención de la base de datos por el simulador T1DMS	21
2.1.1 Ingesta de glucosa	21
2.1.2 Insulina Inyectada	22
2.1.3 Dinámica de Concentración de Glucosa en Sangre	24
2.2 Normalización del conjunto de mediciones	24
2.3 Preparación de los datos en secuencias	26
2.4 Entradas y salidas para entrenar el modelo	27
3. Arquitectura del modelo	29
3.1 Número de capas y celdas LSTM	30
3.2 Funciones de activación	33
3.3 Compilación del modelo	34
3.4 <i>Early Stopping</i>	35
4. Entrenamiento del modelo	38
4.1 Validación cruzada K-Fold	39

5. Evaluación del modelo	41
5.1 Desnormalización	41
5.1 Métrica de error MAE	41
5.2 Métrica de error RMSE	42
6. Resultados Obtenidos	43
7. Análisis de resultados	46
8. Conclusiones	48
9. Recomendaciones	50
Referencias Bibliográficas	51
Apéndices	54

Lista de Tablas

	Pág.
Tabla 1 Resultados de las métricas de error de los 10 modelos personalizados, y el modelo generalizado	43

Lista de Figuras

	Pág.
Figura 1 Funcionamiento de una celda LSTM	20
Figura 2 Conjunto de mediciones del T1DMS, ingestas	22
Figura 3 Conjunto de mediciones del T1DMS, insulina	23
Figura 4 Conjunto de mediciones del T1DMS, glucosa	24
Figura 5 Secuencia de entrada y salida del modelo	27
Figura 6 Episodio de glucosa con las entradas y salidas en escala real	28
Figura 7 Episodio de glucosa con la entradas y salidas normalizadas para entrenar el modelo	29
Figura 8 Arquitectura definida del modelo para realizar las predicciones	31
Figura 9 Early Stopping average loss, loss validation vs epoch	37
Figura 10 Grafica del accuracy vs epoch	38
Figura 11 Validación cruzada K-Fold con 5 pliegues	40
Figura 12 Gráfica de la predicción de la glucosa para el modelo personalizado del paciente adulto 02	44
Figura 13 Gráfica de la predicción de la glucosa para el modelo generalizado	45

Lista de Apéndices

	pág.
Apéndice A. Link al repositorio con el código de python en google colab, los datos y las pruebas personalizadas y generales de los 10 pacientes.	54

Glosario

In silico que se realiza en simulación por computadora.

LSTM Long Short-Term Memory - Memoria larga-corto plazo.

RNN Recurrent Neural Network - Redes neuronales recurrentes.

T1DM Type I Diabetes Mellitus - Diabetes mellitus tipo I. Tipo de diabetes, llamada también insulino dependiente debido a que el paciente requiere de dosificación externa de insulina.

T1DMS Type I Diabetes Mellitus Simulator - Simulador de diabetes mellitus tipo I. Simulador desarrollado en MATLAB que contiene las ecuaciones del modelo desarrollado por Boris Kovatchev cuyos parámetros fueron sintonizados para generar el modelo UVA PADOVA.

Resumen

Título: Predicción de los Niveles de Glucosa en Sangre para Pacientes Diabéticos a Partir de Redes Neuronales Recurrentes Tipo LSTM *

Autor: Ismael Vega Rojas **

Palabras Clave: Glucosa, Predicción, Diabetes, Redes Neuronales, In silico, Uva Padova.

Descripción: En este trabajo de grado se presenta un modelo de estimación de glucosa para pacientes con diabetes mellitus tipo 1, donde se usa el simulador T1DMS (Visentin et al., 2018) con el que se obtiene un conjunto de mediciones de insulina y glucosa para entrenar el modelo de predicción. Para realizar este entrenamiento del modelo se utilizan series de tiempo, tomando como entrada valores pasados de glucosa en sangre e insulina inyectada y generando a la salida instantes de tiempo futuro de glucosa en sangre. Las secuencias de entrada constan de un conjunto de 15 muestras (timesteps) medidas cada minuto, mientras que la secuencia de salida, 210 muestras de glucosa futuras predichas mediante el modelo de estimación que corresponde a la curva completa de un episodio de glucosa después de una ingesta de glucosa por parte del paciente. Con estas secuencias de entradas y salidas se entrenan los modelos de *Recurrent Neural Network* (RNN) tipo LSTM, para 10 pacientes cada uno con un modelo personalizado, usando solo los episodios o curvas de glucosa de cada paciente en cada modelo. Y otro modelo generalizado, es decir usando los episodios o curvas de glucosa de todos los pacientes en un solo modelo. Para estimar el rendimiento del modelo, se usan las métricas de error MAE y RMSE, se evalúa el ajuste del modelo, se define la arquitectura final utilizando el número de capas y celdas que permitan una mejor precisión, al igual que el optimizador que mejor funcione, como los inicializadores adecuados, y las funciones de activación en cada capa del modelo, para que se obtengan las predicciones más exactas. Donde se toma como base lo aprendido en el curso electivo de Deep learning, dictado en el 2022-2, en la Universidad Industrial de Santander.

* Trabajo de Grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: RODOLFO VILLAMIZAR MEJIA. PhD. Tecnologías de la información Codirector: EDWARD ALFONSO RODRIGUEZ MORENO. MSc Ingeniería Electrónica.

Abstract

Title: Prediction of Blood Glucose Levels for Diabetic Patients from LSTM-Type Recurrent Neural Networks*

Author(s): Ismael Vega Rojas**

Key Words: Glucose, Prediction, Diabetes, Neural Networks, In silico, Uva Padova.

Description: In this degree project, a glucose estimation model for patients with type 1 diabetes mellitus is presented, where the T1DMS simulator (Visentin et al., 2018) is used to obtain a set of insulin and glucose measurements to train the prediction model. To carry out this training of the model, time series are used, taking as input past values of blood glucose and injected insulin and generating instants of future time of blood glucose as output. The input sequences consist of a set of 15 samples (timesteps) measured every minute, while the output sequence consists of 210 future glucose samples predicted by the estimation model that corresponds to the complete curve of a glucose episode after a glucose intake by the patient. With these sequences of inputs and outputs, the LSTM-type Recurrent Neural Network (RNN) models are trained for 10 patients each with a personalized model, using only the episodes or glucose curves of each patient in each model. And another generalized model, that is, using the episodes or glucose curves of all the patients in a single model. To estimate the performance of the model, the MAE and RMSE error metrics are used, the fit of the model is evaluated, the final architecture is defined using the number of layers and cells that allow better precision, as well as the optimizer that works best. , such as the proper initializers, and activation functions at each model layer, so that the most accurate predictions are obtained. Where what was learned in the Deep learning elective course, taught in 2022-2, at the Industrial University of Santander.

* Degree Work

** Faculty of Physics and Mechanical Engineering. School of Electronic and Telecommunications Engineering. Director: RODOLFO VILLAMIZAR MEJIA. PhD in Information Technology Co-director: EDWARD ALFONSO RODRIGUEZ MORENO MSc in Engineer Electronic.

Introducción

Para el año 2019, según la organización mundial para la salud (OMS, 2022), la diabetes fue la causa directa de 1,5 millones de defunciones y, de todas las muertes por diabetes, un 48% tuvo lugar antes de los 70 años de edad. Esto representa un incremento importante, ya que en 2014, un 8,5% de los mayores de 18 años padecían diabetes (OMS, 2022).

La Diabetes Mellitus Tipo 1 (T1DM), se debe a la destrucción inmunológica de las células beta pancreáticas produciendo una deficiencia absoluta de insulina (Katsarou, Gudbjörnsdottir, Rawshani, et al., 2017). Esto hace que la glucosa se acumule en la sangre, y no permite que los alimentos que se consumen se conviertan en energía para el cuerpo, afectando la salud y bienestar de las personas que padecen esta enfermedad. Debido a que es una enfermedad que aún no tiene cura, algunas soluciones se han propuesto con el objetivo de contribuir al manejo y bienestar de la persona, siendo la predicción de los niveles de glucosa en sangre, un factor de gran importancia.

La predicción de los niveles de glucosa no sería posible si no se pudiese tener registro de ella, es por esto que gracias al uso de sistemas de monitoreo continuo de glucosa (CGM por sus siglas en inglés) se pueda obtener un conjunto de mediciones, almacenarlas, pre procesarlas y posteriormente se utilicen para entrenar complejos algoritmos de predicción, entre los que se destacan aquellos basados en técnicas de inteligencia artificial y redes neuronales.

Una de las arquitecturas que más se usan en el Machine Learning son las Recurrent Neural Network (RNN), estas son capaces de procesar diferentes tipos de secuencias, y se suelen usar ampliamente para tareas como reconocimiento de voz, análisis de video, o procesamiento del lenguaje natural (Ng, 2018). No obstante, las configuraciones básicas de RNN's, tienen limitantes, ya que cuentan con memorias a corto plazo. En este trabajo de grado, se usan las redes tipo Long-

Short Term Memory (LSTM) para predecir los niveles de glucosa, que son las principales arquitecturas de RNN, ya que son las más robustas y permiten contar con una memoria a más largo plazo (Alammar, 2021).

Las características de las RNN tipo LSTM, las hacen adecuadas para que se creen modelos predictivos de los niveles de glucosa en la sangre para pacientes con T1DM. Para entrenar la red neuronal, se usa una generación de un conjunto de mediciones que se obtienen del simulador The UVA/PADOVA Type 1 Diabetes Simulator (T1DMS).

Con la predicción se puede tener una estimación de glucosa ingerida y así se generan referencias similares a las de los sujetos sanos, que mediante un sistema de dosificación automática de insulina (páncreas artificial) se siga el perfil de glucosa deseado, con el uso de los datos del simulador T1DMS acordes a situaciones reales y permitiendo un apropiado entrenamiento de las RNN tipo LSTM. Logrando de esta manera, un impacto positivo en el estilo de vida de las personas con T1DM, que podrán seguir un perfil de glucosa de personas sanas y evitar episodios de hiperglucemia o hipoglucemia.

Objetivos

Objetivo General

Predecir los niveles de glucosa en sangre para pacientes con DM1, in silico, a partir del desarrollo de redes neuronales recurrentes tipo LSTM.

Objetivos Específicos

Definir la arquitectura de la red neuronal recurrente tipo LSTM, en función de las condiciones de entrada.

Entrenar la red neuronal recurrente tipo LSTM.

Validar in silico la red entrenada evaluando su desempeño con métricas de error como RMSE y MAE

1. Marco conceptual

1.1 Diabetes

La diabetes es una enfermedad crónica que afecta a millones de personas en todo el mundo. Según la Organización Mundial de la Salud (OMS, 2022), la diabetes es la séptima causa de muerte en el mundo y se estima que en 2021 había alrededor de 422 millones de adultos con diabetes en todo el mundo (OMS, 2022).

La diabetes se caracteriza por niveles elevados de glucemia (azúcar en la sangre) debido a una falta de producción de insulina por el páncreas (diabetes tipo 1) o a una resistencia a la insulina por parte del cuerpo (diabetes tipo 2). También existen otros tipos de diabetes, como la diabetes gestacional y la diabetes relacionada con otros trastornos (OMS, 2022).

Entre los factores de riesgo de la diabetes se incluyen la obesidad, la falta de actividad física, la edad, el historial familiar de diabetes, la hipertensión arterial y la mala alimentación. La diabetes puede causar complicaciones a largo plazo, como enfermedades cardiovasculares, daño renal, pérdida de visión y amputación de miembros (OMS, 2022).

1.1.1 Diabetes mellitus tipo 1 (T1DM)

La diabetes mellitus tipo 1 es un trastorno metabólico crónico que se caracteriza por la deficiencia de insulina debido a la pérdida de células beta por parte del páncreas (Katsarou, Gudbjörnsdottir, Rawshani, et al., 2017). La insulina es una hormona que regula el nivel de glucosa en la sangre, permitiendo que la glucosa entre en las células para ser utilizada como energía. Hoy por hoy aún se desconoce la causa de este tipo de diabetes y tampoco se sabe cómo prevenirla (OMS, 2022).

Según la Organización Mundial de la Salud (OMS, 2022), la diabetes tipo 1 es responsable del 5-10% de todos los casos de diabetes en todo el mundo, y se estima que alrededor de 1.1 millones de niños y adolescentes menores de 20 años tienen esta enfermedad.

1.1.1 Modelo Uva Padova

Es el modelo que usa el simulador metabólico de diabetes mellitus tipo I (T1DMS) desarrollado por el hospital Universitario de Padova en Italia y la Universidad de Virginia, en Estados Unidos. Este se basa en las ecuaciones de la dinámica de la glucosa en la sangre, la cinética de la insulina y el glucagón, así como en la respuesta fisiológica del cuerpo a la ingesta de alimentos, la actividad física y otros factores. El T1DMS proporciona a los pacientes con diabetes mellitus tipo 1 y a los profesionales de la salud una herramienta para simular los efectos de diferentes terapias y estrategias de manejo de la diabetes en el control de la glucemia.

El modelo Uva Padova se basa en un enfoque multidisciplinario que involucra a endocrinólogos, diabetólogos, enfermeros, psicólogos, nutricionistas y otros profesionales de la salud para brindar atención personalizada y adaptada a las necesidades individuales de cada paciente. Este modelo se basa en un conjunto de ecuaciones que describen la dinámica del metabolismo de la glucosa en el cuerpo humano.

El modelo Uva Padova ha demostrado ser efectivo en mejorar el control de la glucemia y reducir las complicaciones de la diabetes mellitus tipo 1 en estudios clínicos y en la práctica clínica (Visentin et al., 2018).

1.2 Redes Neuronales Recurrentes tipo LSTM

Una red LSTM (Long Short-Term Memory) es un tipo de arquitectura de red neuronal recurrente (RNN) que se utiliza en el campo del deep learning. El deep learning es una rama del

aprendizaje automático que se enfoca en entrenar modelos de redes neuronales profundas para aprender representaciones y características complejas de los datos (Ng, 2018).

Estos modelos se utilizan para resolver problemas de aprendizaje automático, tales como la clasificación de imágenes, la traducción de idiomas, la detección de objetos y el procesamiento del lenguaje natural (Géron, 2019).

También se han llevado a cabo estudios sobre la predicción de los niveles de glucosa en sangre utilizando una red neuronal recurrente tipo LSTM, como en los estudios hechos por Katsarou, Geroga, Christou, Tigas, Papaloukas y Fotiadis (2022), donde el objetivo es la predicción personalizada de glucosa a corto plazo, con modelos probados en 29 pacientes, logrando un desempeño con métricas RMSE de 3.13, 6.41 y 8.81 mg/dL y MAE de 1.98, 5.06 y 6.47 mg/dL para un horizonte de predicción de 5, 15 y 30 minutos.

Algunos estudios evalúan diferentes redes neuronales como las redes neuronales recurrentes (RNN), las unidades recurrentes cerradas (GRU) y las memorias a corto-largo plazo (LSTM), como en los estudios realizados por Celik & Varli (2022), donde el horizonte de pronóstico de glucosa en sangre se decide en 30 minutos, con pruebas hechas a 12 pacientes, y los métodos son evaluados utilizando las métricas de error RMSE y MAE.

Otros estudios de modelos predictivos con redes LSTM para pacientes diabéticos tipo 1, son los realizados por Iacono, Magni & Toffanin (2022), donde el Páncreas Artificial incluye controladores, como el control predictivo de los modelos, para mantener los niveles de glucosa en la sangre bajo control. En el que se proponen modelos LSTM con un horizonte de predicción de 40 minutos, considerando la insulina y los valores de glucosa en sangre como entrada, donde se obtuvo un resultado en la métrica de error RMSE de 7.67 mg/dL.

Para la predicción personalizada de glucosa en sangre usando dos capas ocultas LSTM con dos capas densas, de los estudios realizados por Shuvo & Islam (2023), donde se evalúa el desempeño del modelo con métricas de error RMSE de 16.06 mg/dL y MAE de 10.64 mg/dL, con un horizonte de predicción de hasta 120 minutos.

Con todos estos estudios se puede tener una idea de lo que puede hacer la red LSTM en las series de tiempo para la predicción de glucosa, y por esto sea la capa oculta que se use en la arquitectura del modelo de este trabajo de grado, como también es importante entender el funcionamiento de una celda LSTM.

La red LSTM es capaz de recordar un dato relevante, en la secuencia y de preservarlo por varios instantes de tiempo, por tanto, puede tener una memoria tanto de corto plazo, como las redes neuronales recurrentes, como también de largo plazo. La red LSTM está en la capacidad de añadir o eliminar la información que considere relevante para el procesamiento de la secuencia (Codificando bits, 2019).

Según Codificando bits (2019), en el video “¿Qué es una red LSTM?”, se explica a continuación el funcionamiento de las celdas LSTM en el procesamiento de secuencias. La celda LSTM tiene una entrada y una salida adicional, este elemento adicional se conoce como celda de estado (cell state) que es la clave del funcionamiento de las redes LSTM, la celda de estado es como una banda transportadora a la que se puede añadir o remover datos que no se quiere que queden en la memoria de la red, para eso se usan varias compuertas como se puede ver en la Figura 1, la compuerta forget gate que permite eliminar elementos de la memoria, también la compuerta update gate que permite añadir nuevos elementos a la memoria, y el output gate que es la compuerta de salida que permite crear el estado oculto actualizado (Codificando bits, 2019).

Estas compuertas son redes neuronales que funcionan como válvulas, totalmente abiertas permiten el paso de información, y totalmente cerradas lo bloquean por completo. Cada una de estas compuertas está conformada por tres elementos, una red neuronal, una función sigmoïdal, y un elemento multiplicador.

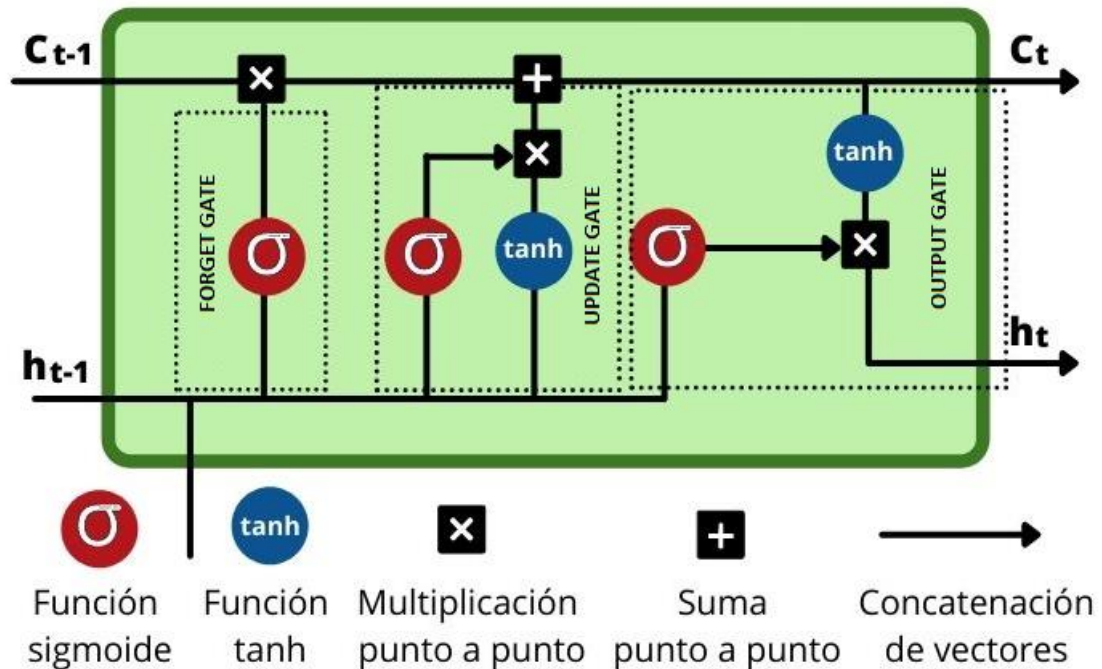
El uso de las compuertas, permite manipular la memoria de la celda LSTM, la compuerta forget gate, permite decidir que información se va descartar, por tanto no pasará a la celda de estado, para ello se toma el estado oculto anterior y la entrada actual, los transforma y los lleva a la función de activación sigmoïdal, los pesos se aprenden durante el entrenamiento, y como salida genera un vector, si uno de los valores de este vector es cero, o cercano a cero, entonces la celda LSTM eliminara esa porción de información, mientras que si alcanza valores iguales o cercanos a uno, esta información se mantendrá y llegará a la celda de estado. El update gate, permite actualizar la memoria de la celda LSTM, para ello se toma el estado oculto anterior, y la entrada actual, se transforman, y se lleva de nuevo a una función de activación sigmoïdal, también en este caso los pesos se aprenden durante el entrenamiento, y como salida de esta compuerta genera otro vector.

Los valores que se quiere preservar en la memoria de la red, serán aquellos cercanos a uno, se tienen ya los datos generados por las compuertas forget y update, y ahora se actualiza la celda de estado, es decir la memoria de la red LSTM. En primer lugar se elimina la información irrelevante de la celda de estado, se multiplica el valor anterior de esta celda, por el vector generado por la compuerta forget, a continuación se forma un vector de valores candidatos a formar parte de la nueva memoria, de nuevo los parámetros w y b se aprenden durante el entrenamiento, ahora se filtran estos valores, multiplicando punto a punto el vector que se acaba de obtener, con el generado con la compuerta update y el resultado se usa a los valores anteriores de la celda de estado, creando así la memoria actualizada.

Finalmente se calcula el nuevo estado oculto, para lo cual se usa el output gate o compuerta de salida, este estado oculto de salida, es simplemente una versión filtrada de la celda de estado que se acaba de generar, en primer lugar se escala el nuevo cell state, para garantizar que esté en el rango de -1 a 1, el rango que tiene precisamente el estado oculto, para ello se usa la función tangente hiperbólica, ahora la compuerta de salida, para que se determina las porciones del cell state que entrarán a formar parte del nuevo estado oculto, al igual que en los casos anteriores, los parámetros w y b se aprenden durante el entrenamiento, y finalmente se filtran los valores de cell state con el vector generado por la compuerta de salida, con esto ya tenemos todos los elementos que conforman una red LSTM.

Figura 1

Funcionamiento de una celda LSTM



Nota. La imagen proporciona una visión general de la arquitectura de la red LSTM (Long Short-Term Memory). Tomado de Abdatum (Recuperado en 2023 de <https://abdatum.com/tecnologia/redes-neuronales-recurrentes>)

2. Conjunto de datos

2.1 Obtención de la base de datos por el simulador T1DMS

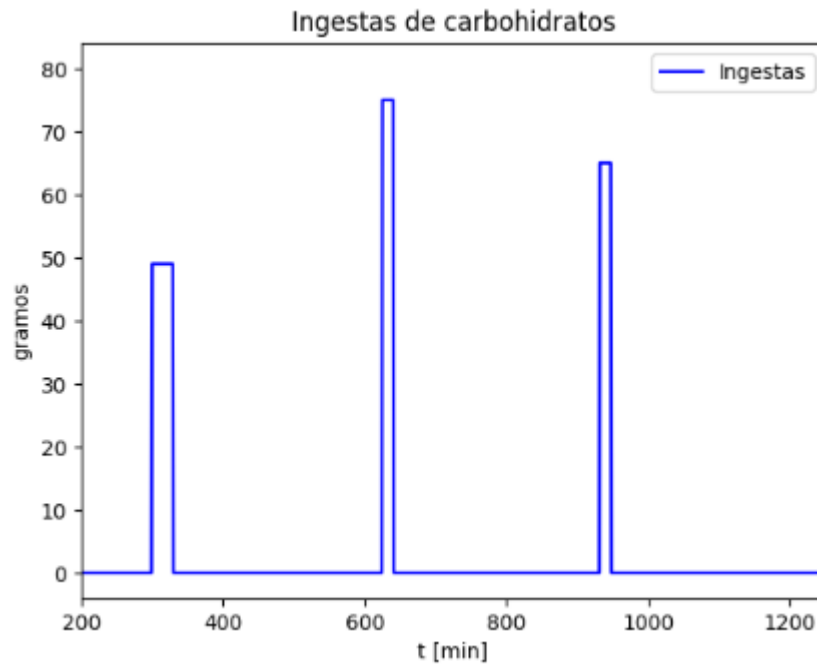
Para obtener la base de datos, se usa el simulador The UVA/PADOVA Type 1 Diabetes Simulator (T1DMS) de Simulink/MATLAB, donde se programa la dinámica, con un cronograma de 3 comidas al día ricas en carbohidratos, una dosis de insulina de acción lenta que actúa durante todo el día, y dosis de insulina de acción rápida en los momentos de ingesta de comida en función de la cantidad de carbohidratos que se consume. Bajo estas condiciones, se obtuvo una base de 10 sujetos adultos in silico, durante 20 días donde se registra la dinámica de la glucosa en sangre. Por tanto, mediante el simulador T1DMS se obtienen las ingestas de comida, la insulina aplicada diaria, así como la dinámica de glucosa en sangre, para cada uno de los 10 pacientes.

2.1.1 Ingesta de glucosa

Las ingestas simuladas corresponden a 3 por cada día, es decir lo que consume el paciente a la hora del desayuno en un rango de 45 -50 gramos de carbohidratos, al almuerzo en un rango de 75 - 150 gramos de carbohidratos, y la cena en un rango de 65 - 70 gramos de carbohidratos. En la Figura 2 se representan las mediciones de las ingestas, obtenido del simulador T1DMS, del paciente adulto 02, son las ingestas de carbohidratos en gramos (g) para el día 1.

Figura 2

Conjunto de mediciones del T1DMS, ingestas



2.1.2 Insulina Inyectada

En un paciente diabético controlado, normalmente se inyecta una insulina basal diaria, que tiene un efecto lento durante todo el día, y mantiene un nivel basal de insulina, y en los momentos donde hay una ingesta, se aplica otra insulina de acción rápida (denominados bolos de insulina), los llamados bolos de insulina, dependiendo la cantidad de carbohidratos que se consumen. La insulina de acción rápida, se aplica para que se reduzcan más rápidamente los niveles de glucosa antes de comenzar la siguiente ingesta.

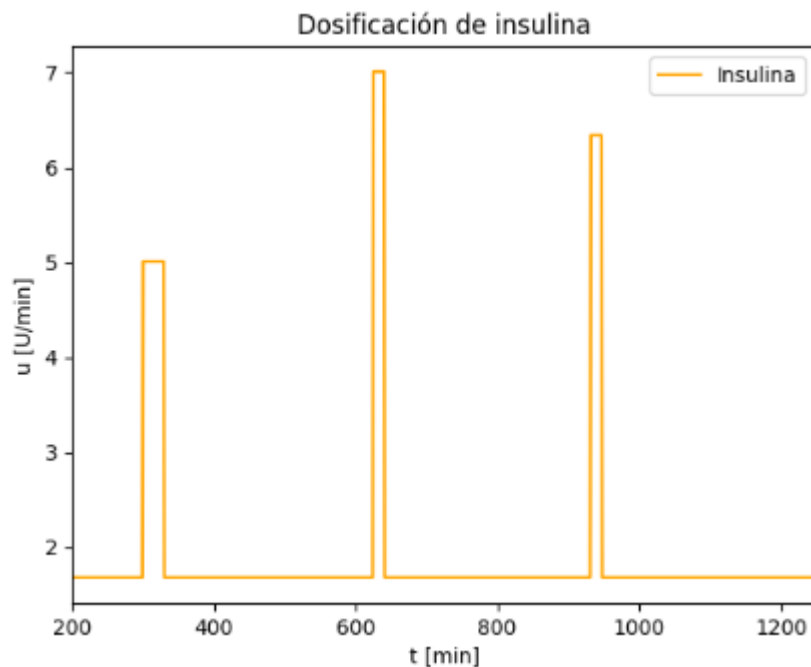
Para calcular la dosis ideal de insulina de cobertura de carbohidratos se utiliza una fórmula que relaciona la equivalencia entre insulina y carbohidratos; donde se establece la cantidad de carbohidratos que cubre una unidad de insulina y suele ser entre los 10 y 15 gramos de carbohidratos. Así que se toma, el valor máximo del rango de los carbohidratos a consumir, para

cada una de las ingestas, para el desayuno el valor de la insulina de acción rápida será de 3.33 unidades, para el almuerzo de 5.33 unidades, y para la cena de 4.66 unidades de insulina.

Cabe recordar que estos valores de insulina de acción rápida en la simulación se escogen en un rango permitido, para evitar que los niveles de glucosa se eleven demasiado, más no para corregir el comportamiento de la glucosa. Y que en las condiciones reales de un paciente, resulta difícil saber cuántos gramos de glucosa consume, y este valor de insulina de acción rápida es usualmente un cómputo manual, en base a una dieta estricta. Y lo que busca el páncreas artificial es mantener los niveles de glucosa en una referencia ya establecida, pero no sigue el comportamiento de glucosa de un sujeto sano, y esto último es lo que se busca con el modelo predictivo. En la Figura 3 se representan las mediciones de las unidades de insulina, para un día, obtenidas del T1DMS, del paciente adulto 02.

Figura 3

Conjunto de mediciones del T1DMS, insulina

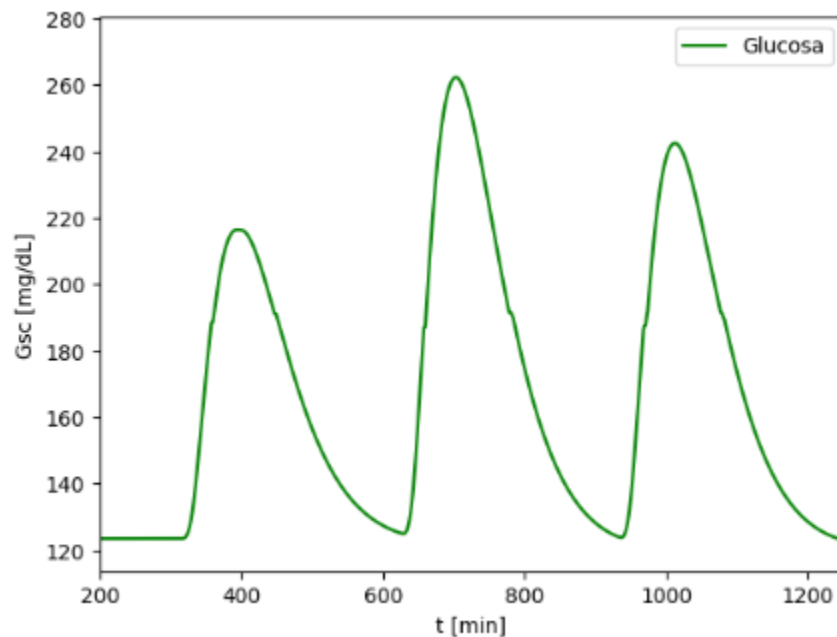


2.1.3 Dinámica de Concentración de Glucosa en Sangre

La dinámica de concentración de glucosa en sangre, son las mediciones generadas por el modelo Uva Padova en el simulador T1DMS, que dependiendo de los gramos de carbohidratos y de unidades de insulina, va a tener mayor, o menor cantidad de glucosa en la sangre. En la Figura 4 se representa las mediciones del comportamiento de glucosa en sangre, en miligramos por decilitro (mg/dL), obtenido del T1DMS, para un día con tres ingestas (desayuno, almuerzo y comida), obtenidas del T1DMS, del paciente adulto 02 y cada curva representa un episodio de glucosa.

Figura 4

Conjunto de mediciones del T1DMS, glucosa



2.2 Normalización del conjunto de mediciones

La normalización de los datos de entrada es una técnica comúnmente utilizada en el procesamiento de datos para mejorar el rendimiento de los modelos de redes neuronales (Ng,

2018), incluyendo el modelo de RNN tipo LSTM utilizado para predecir los valores de glucosa en la sangre. Normalizar los datos implica transformar los valores de entrada para que tengan una escala común, lo que puede ayudar a la red neuronal a converger más rápidamente durante el entrenamiento y mejorar la precisión de las predicciones.

De acuerdo con De Faria (2022), se pueden usar técnicas de normalización como `StandardScaler` y `MinMaxScaler` para encontrar patrones que predicen la aparición de hiperglucemia e hipoglucemia usando redes LSTM.

La técnica `MinMaxScaler` escala los valores en un rango de 0 a 1, mientras que la técnica `StandardScaler` estandariza los datos restando la media y dividiendo por la desviación estándar, lo que implica que cada característica tenga una media igual a cero y una desviación estándar igual a uno. Estos valores tendrán una distribución alrededor de cero, con algunos valores positivos y otros negativos (De Faria, 2022).

En este trabajo de grado se normaliza con `StandardScaler`, ya que se tiene en la entrada dos características de insulina y glucosa, y al estar todo en la misma escala, hace que la correlación prevalezca y se asignen bien los pesos. Esto debido a que la glucosa en sangre tiene valores mucho más grandes en comparación con la insulina inyectada y al escalar sus valores, permiten que tengan una influencia equilibrada en el modelo. Así que no hay un límite específico para los valores normalizados, ya que dependen de cómo se distribuyen los valores originales.

Al ajustar `StandardScaler` en los datos de entrenamiento con `sc.fit_transform(X_train)`, se está calculando la media y la desviación estándar de los datos de entrenamiento y aplicándolas para normalizar tanto los datos de entrenamiento como los datos de prueba. Luego, al utilizar `sc.transform(X_test)`, se está aplicando la misma transformación a los datos de prueba utilizando la media y desviación estándar calculadas en los datos de entrenamiento.

2.3 Preparación de los datos en secuencias

Preparar los datos en secuencias es una técnica comúnmente utilizada en el aprendizaje profundo y en particular en el entrenamiento de redes neuronales recurrentes, como LSTM, para predecir valores futuros (Alammar, 2021). En la predicción de glucosa, los datos se organizan en secuencias para que el modelo pueda aprender a predecir valores de glucosa futuros en función de los valores anteriores.

Lo que se hace es preparar los datos de entrenamiento y prueba, que consiste en series de tiempo de entrada “X” y salida “y” del modelo. Las salidas “y” serían la respuesta verdadera a la cual se quiere llegar (ground truth).

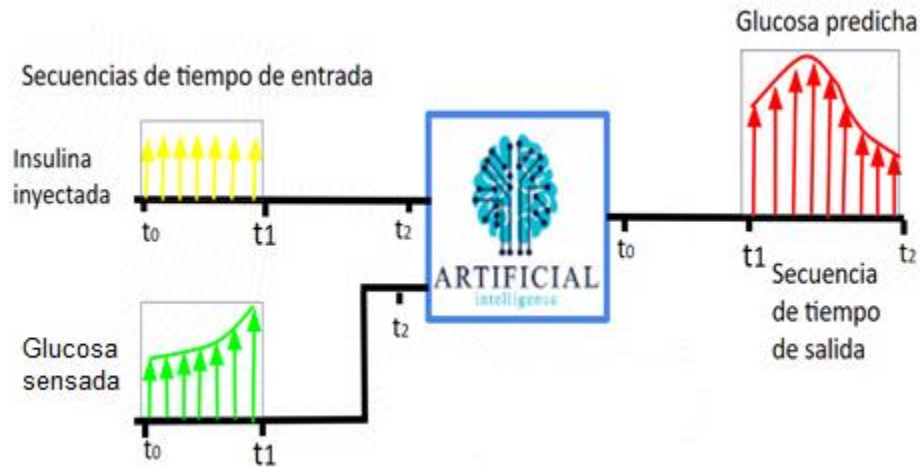
La idea detrás de esto es que, en lugar de entrenar el modelo para predecir un solo valor de glucosa, el modelo aprende a predecir una secuencia de valores de glucosa. De esta manera, el modelo puede capturar mejor las relaciones, y patrones en los datos de glucosa a lo largo del tiempo.

Como se puede ver en la Figura 5, los datos se separan en secuencias de tiempo, para entrenar el modelo de redes neuronales recurrentes tipo LSTM, y poder predecir los valores de glucosa, con entradas de insulina inyectada y valores de glucosa sensado.

Mientras tanto, la salida para entrenar el modelo, también se separa en secuencias de tiempo, con las muestras futuras de glucosa que se quieren predecir, donde las pruebas se realizan con tamaños de ventanas mayores a las de entrada, estimando así más muestras de salida, para cubrir la curva del episodio de glucosa que se quiere predecir.

Figura 5

Secuencia de entrada y salida del modelo



2.4 Entradas y salidas para entrenar el modelo

Para entrenar el modelo se usan secuencias de entrada de 15 muestras, que corresponden a los valores actuales y anteriores de glucosa e insulina, es decir los 15 minutos que empiezan a generar el episodio o curva de glucosa en la sangre, posterior a la glucosa ingerida por el paciente. Un tiempo mayor no es apropiado porque lo importante es que se pueda estimar la ingesta de glucosa lo más rápido posible para que en el montaje del páncreas artificial se pueda generar una referencia de curva de glucosa en función de dicha ingesta, tal que se intente seguir el comportamiento de glucosa de un sujeto sano.

Los valores de glucosa que se quieren predecir, son los de un episodio o curva de glucosa que permita calcular la ingesta ingerida, así que el tamaño de la secuencia de salida es de 210 muestras, que corresponden a los valores siguientes de glucosa, es de decir la predicción corresponde a 3 horas y 30 minutos, ya que la frecuencia de muestreo, es de una muestra por minuto.

En la Figura 6 se puede ver un episodio de glucosa en escala real, en verde la glucosa de entrada en miligramos por decilitro [mg/dL], en amarillo la entrada de insulina en unidades de insulina [U/min], y en rojo la salida de glucosa con los valores siguientes que se quieren predecir. En la Figura 7 se encuentra el episodio de glucosa con los valores normalizados de las entradas y salidas con los que se va a entrenar el modelo.

Así que el modelo solo se va a entrenar con estos episodios de glucosa de los pacientes con diabetes, donde un 70% de estos episodios van a hacer para entrenamiento, y un 30% para los datos de prueba que no va a ver el modelo durante el entrenamiento.

Figura 6

Episodio de glucosa con las entradas y salidas en escala real

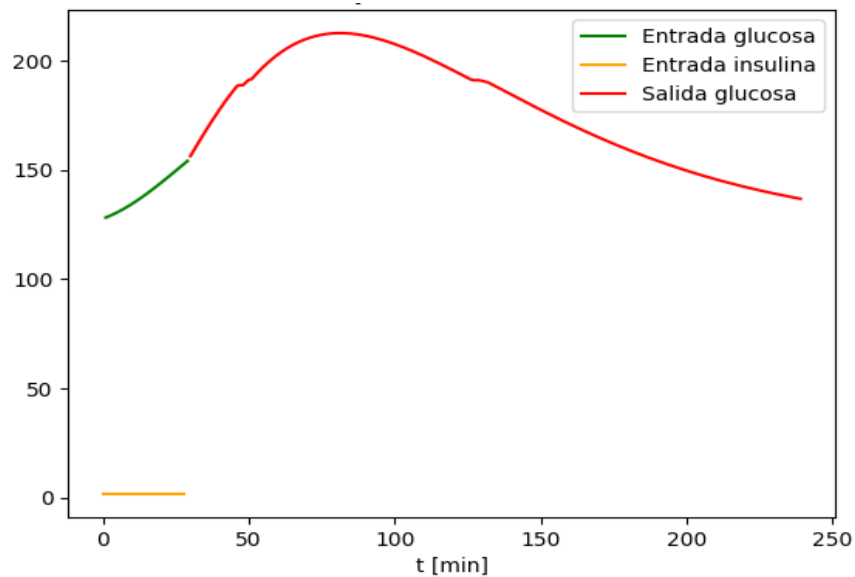
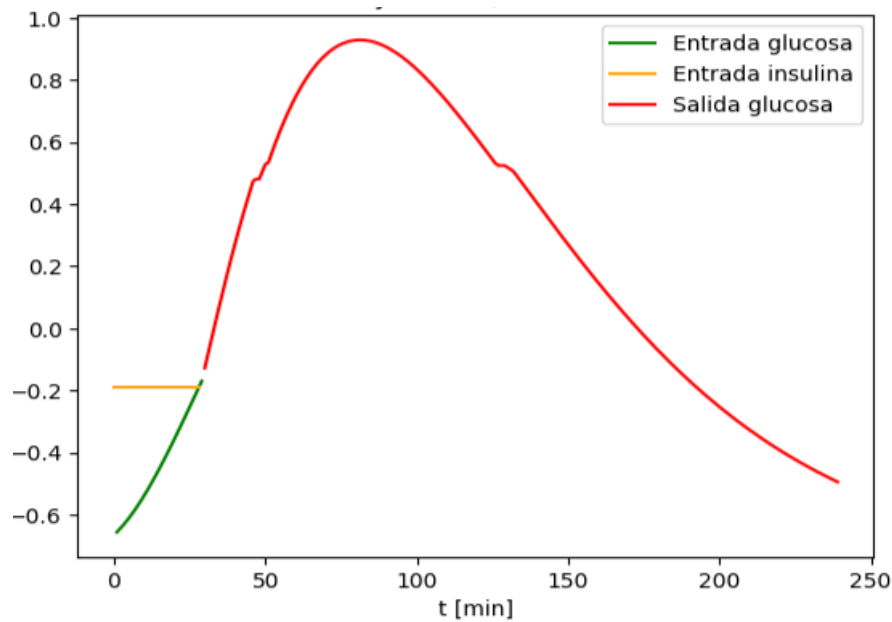


Figura 7

Episodio de glucosa con la entradas y salidas normalizadas para entrenar el modelo



3. Arquitectura del modelo

La estructura de una RNN tipo LSTM consta de una capa de entrada, una o más capas LSTM y una capa de salida (Alammar, 2021). Cada capa LSTM contiene múltiples celdas LSTM, cada una de las cuales consta de tres puertas (input, output y forget gate) y una celda de memoria. Las puertas de entrada y olvido controlan la cantidad de información que se introduce o se olvida en la celda de memoria, mientras que la puerta de salida controla la cantidad de información que se extrae de la celda de memoria.

Las redes LSTM pueden manejar secuencias de datos de longitud variable, y aprender a retener información relevante de las entradas anteriores, para su uso en las siguientes predicciones, donde la información de contexto es importante para la precisión del modelo.

3.1 Número de capas y celdas LSTM

La adición de capas y celdas al modelo puede aumentar la capacidad de la red para aprender patrones complejos en los datos, pero también puede aumentar el riesgo de sobreajuste y la complejidad computacional (Ng, 2018). Por lo tanto, es importante encontrar un equilibrio entre la capacidad de aprendizaje y la eficiencia computacional del modelo al elegir el número de capas adecuado.

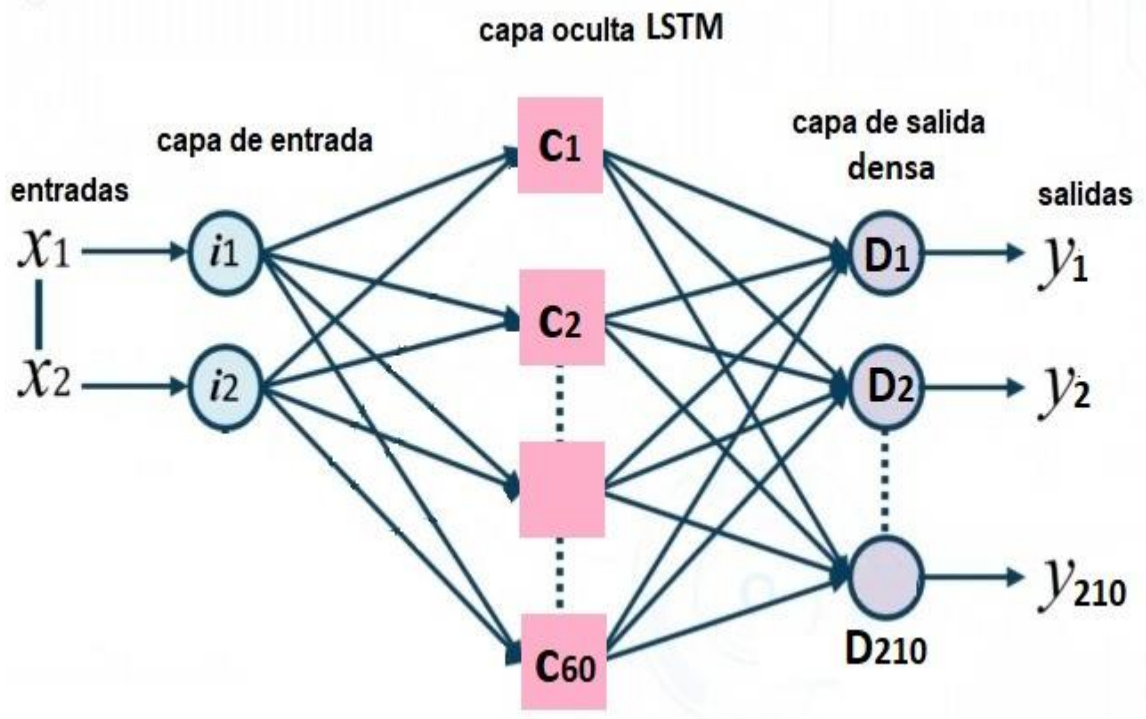
Para determinar si se debe aumentar el número de celdas y capas en el modelo RNN tipo LSTM, es importante monitorear la función de pérdida de error medio cuadrático (MSE) durante el entrenamiento, ya que se espera que disminuya a medida que el modelo aprende a hacer predicciones más precisas. Si la función de pérdida comienza a aumentar, puede ser una señal de que el modelo no está aprendiendo correctamente y puede ser necesario aumentar el número de celdas y capas para mejorar su capacidad de aprendizaje.

El error de validación, por otro lado, mide el rendimiento del modelo en un conjunto de datos que no se utilizó para entrenarlo. Durante el entrenamiento, el modelo se evalúa en el conjunto de validación para determinar si está aprendiendo bien y no está sobreajustando los datos de entrenamiento. Si el error de validación comienza a aumentar o no mejora, puede ser una señal de que el modelo no está aprendiendo bien y puede ser necesario aumentar el número de celdas y capas para mejorar su capacidad de generalización.

En la Figura 8, se puede ver la arquitectura final del modelo que se escogió para realizar las predicciones de la glucosa en los 10 pacientes T1DM. Este modelo tiene una capa de entrada de un tamaño de secuencia de 15, con dos características, una de insulina y otra de glucosa, la capa oculta LSTM tiene 60 celdas, y la salida es una capa densa de 210 neuronas. Esta arquitectura de RNN consta de una capa LSTM y una capa densa para regresión. La capa LSTM se encarga de procesar las secuencias de entrada a través de conexiones recurrentes que permiten que la información fluya de una iteración a la siguiente. Después de la capa LSTM, se agrega una capa densa. La capa densa toma la salida de la capa LSTM y realiza una regresión lineal en ella, generando la predicción.

Figura 8

Arquitectura definida del modelo para realizar las predicciones



Nota. La imagen proporciona una visión general de la arquitectura usada en el modelo de predicción de glucosa, que tiene una capa de entrada con dos características de insulina inyectada y glucosa sensada, con una capa oculta LSTM de 60 celdas, y una capa densa de 210 neuronas. Tomado de Análisis y Programación OOP (Recuperado en 2023 de <https://analisisyprogramacionoop.blogspot.com/2020/05/introduccion-redes-neuronales.html>)

Se realizan diferentes pruebas para escoger el número de celdas y de capas, al paciente adulto 10. Con 1 capa y 100 celdas se obtuvo un error MAE de 56.7385, también un error RMSE de 62.1921 en las predicciones, al igual con 1 capa y 32 celdas se obtuvo errores MAE de 39.0776 y RMSE de 39.7106 en las predicciones, con 2 capas se obtuvo un error MAE de 40.7999 y RMSE de 46.2494. Mientras que con 1 capa y 60 celdas se obtuvo error MAE de 6.3796 y RMSE de 9.4083, por lo tanto, se escogió este último número de capas y de celdas.

En la capa oculta LSTM y en la capa de salida densa, se usaron los inicializadores que se utilizan para proporcionar una forma inicial de los pesos y las celdas de memoria que permitan a la red aprender eficientemente a partir de los datos de entrada. El inicializador `he_uniform` tuvo mejores resultados en las métricas de error MAE y RMSE en comparación que otros inicializadores como `glorot_uniform`, y `random_uniform`, al igual que mejores resultados en la validación MSE durante el entrenamiento.

El tamaño del lote o batch size se refiere al número de muestras de entrenamiento que se utilizan en una iteración de entrenamiento de un modelo de red neuronal. Durante el entrenamiento, los datos de entrada se dividen en lotes y se utilizan para actualizar los pesos del modelo después de cada lote. En este caso se utilizan lotes de tamaño 32 para el entrenamiento del modelo.

Un tamaño de lote más grande puede permitir un procesamiento más rápido de los datos de entrenamiento en paralelo, pero también puede requerir más memoria para almacenar los cálculos intermedios durante la propagación hacia atrás. Por otro lado, un tamaño de lote más pequeño puede requerir menos memoria, pero puede ser más lento en términos de procesamiento de datos. Así que se ajusta el tamaño del lote adecuadamente según el conjunto de datos y el modelo para lograr un equilibrio entre la eficiencia y la precisión del modelo.

También se realizan pruebas con optimizadores, que son utilizados para ajustar los pesos y los sesgos que minimicen la función de pérdida. Se prueban los optimizadores SGD, Adam, y el RMSprop, con este último se obtuvo mejores resultados en la validación de la función de pérdida MSE durante el entrenamiento, y en las métricas de error MAE y RMSE, por lo tanto se escoge al optimizador RMSprop como un hiperparámetro del modelo.

3.2 Funciones de activación

Para una RNN tipo LSTM que hace predicciones de los valores de glucosa para pacientes diabéticos, hay varias opciones de funciones de activación que se pueden usar para la capa de salida y las capas intermedias de la red neuronal.

Para la capa de salida, la función de activación comúnmente utilizada para la regresión es la función lineal, ya que permite la predicción de valores continuos sin restricciones (Ng, 2018). La función lineal no comprime los valores de salida y, por lo tanto, puede predecir cualquier valor real, positivo o negativo. Por lo tanto, la capa densa de salida tiene una función de activación lineal.

En cuanto a la capa intermedia, se utiliza una función de activación tangente hiperbólica (\tanh), que es una función matemática utilizada en redes neuronales para introducir no linealidad en la salida de las capas ocultas de la red. La función tangente hiperbólica toma cualquier número

como entrada y produce un número entre -1 y 1 como salida, lo que le hace útil para normalizar valores entre un rango determinado. La fórmula de la función tangente hiperbólica es la siguiente:

$$\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (1)$$

Donde "e" es la constante matemática e (aproximadamente 2.71828) y "x" es la entrada.

En una red neuronal, la función de activación determina si una unidad neuronal se activa o no y qué valores se propagan a través de la red. La función de activación tangente hiperbólica (tanh) es una opción popular como función de activación para las unidades LSTM y otras capas porque es simétrica alrededor de cero y trata los valores positivos y negativos de manera similar.

Se probaron otras funciones de activación en la capa oculta LSTM, como la función de activación de Unidad Lineal Rectificada (ReLU), pero tenía valores más altos en las métricas de error RMSE de 6.2337, como también se probó la función de activación Unidad Lineal Rectificada con Fuga (Leaky ReLU), con un valor RMSE de 5.935, que no representaba una mejora importante con respecto a tanh que tuvo un error RMSE de 5.3687 por lo que se dejó la función de activación tangente hiperbólica.

3.3 Compilación del modelo

Una vez que se han agregado las capas, el modelo se compila con el optimizador RMSprop y la función de pérdida de error cuadrático medio (por sus siglas en inglés MSE). El optimizador RMSprop es un algoritmo comúnmente utilizado para optimizar redes neuronales y la función de pérdida MSE es común en problemas de regresión, ya que mide la diferencia cuadrática media entre las predicciones y los valores verdaderos.

La fórmula para calcular el MSE es la siguiente:

$$MSE = \frac{1}{n} * \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

Donde:

- n es el número de muestras en los datos de entrenamiento
- y_i es el valor real de la i -ésima muestra
- \hat{y}_i es el valor predicho por el modelo para la i -ésima muestra

La función MSE penaliza de manera más significativa las predicciones que se desvían significativamente de los valores reales, ya que eleva al cuadrado las diferencias entre las predicciones y los valores reales. Esto significa que los errores más grandes tienen un mayor impacto en el valor total de la función de pérdida.

El optimizador RMSprop se adapta la tasa de aprendizaje, pero a diferencia de Adam, solo tiene en cuenta los momentos de segundo orden del gradiente. Y a diferencia del SGD, que utiliza una tasa de aprendizaje constante para todos los parámetros, RMSprop ajusta la tasa de aprendizaje de manera individual para cada parámetro (Zou, F., Shen, L., Jie, Z., Zhang, W., & Liu, W., 2019).

En cada iteración, RMSprop calcula una estimación del segundo momento del gradiente acumulado utilizando un promedio ponderado exponencialmente decaiente. Esta estimación se utiliza para ajustar el learning rate de cada parámetro antes de realizar las actualizaciones (Zou, F., Shen, L., Jie, Z., Zhang, W., & Liu, W., 2019).

3.4 *Early Stopping*

El *Early Stopping* es una técnica que se utiliza para evitar el sobreajuste en los modelos de aprendizaje automático. Consiste en detener el entrenamiento del modelo antes de que se complete todo el número de épocas definido, en función de una condición establecida previamente, como la disminución de la pérdida en un conjunto de validación o la mejora en una métrica de rendimiento específica (Liang, H., Zhang, S., Sun, J., He, X., Huang, W., Zhuang, K., & Li, Z., 2019).

Este es un hiperparámetro porque se debe elegir cuándo detener el entrenamiento del modelo, y qué condición se debe cumplir para detenerlo. Los valores de estos parámetros pueden afectar el rendimiento del modelo, ya que detener el entrenamiento demasiado temprano puede hacer que el modelo sea poco preciso, mientras que detener el entrenamiento demasiado tarde puede sobre ajustarlo. Por lo tanto, la elección correcta de los hiperparámetros Early Stopping es esencial para obtener un modelo bien ajustado y generalizado.

En el modelo se utiliza una paciencia de 3, que se refiere al número de épocas adicionales que se permiten entrenar después de que se alcanza el punto de detención, o sea el punto en el que la métrica de validación deja de mejorar. En otras palabras, la paciencia establece un límite para la cantidad de épocas que el modelo puede continuar entrenando sin mejorar en la métrica de validación antes de detener el entrenamiento.

Una paciencia más alta puede permitir una convergencia más completa del modelo, pero también puede aumentar el riesgo de sobreajuste. Por otro lado, una paciencia más baja puede detener el entrenamiento antes de que el modelo alcance su máximo potencial de rendimiento.

La Figura 9 representa como cambia la función de pérdida MSE del entrenamiento y validación a medida que el modelo entrena con más épocas, que ayuda a visualizar cuándo se detiene el entrenamiento de la validación cruzada K Fold, para el paciente adulto 02.

En la Figura 10 se muestra la gráfica del accuracy que ayuda a ver la evolución del modelo por época, donde se muestra como mejora la precisión del modelo a medida que se entrena con más épocas, donde se compara las curvas de precisión en entrenamiento y evaluación. Y la precisión en evaluación nos da una idea de qué tan bien generaliza el modelo a datos no vistos, lo ideal es que la precisión en evaluación aumente a medida que aumentan las épocas, y la precisión en entrenamiento no sea significativamente mayor para evitar el sobreajuste.

Figura 9

Early Stopping average loss, loss validation vs epoch

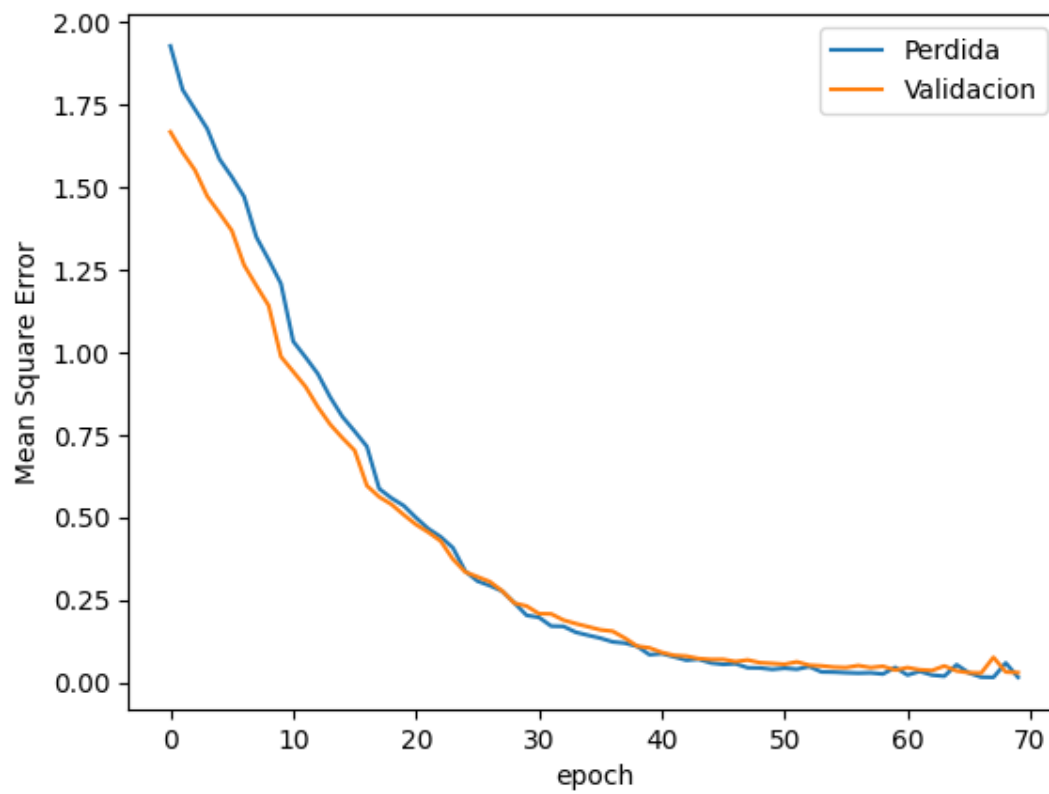
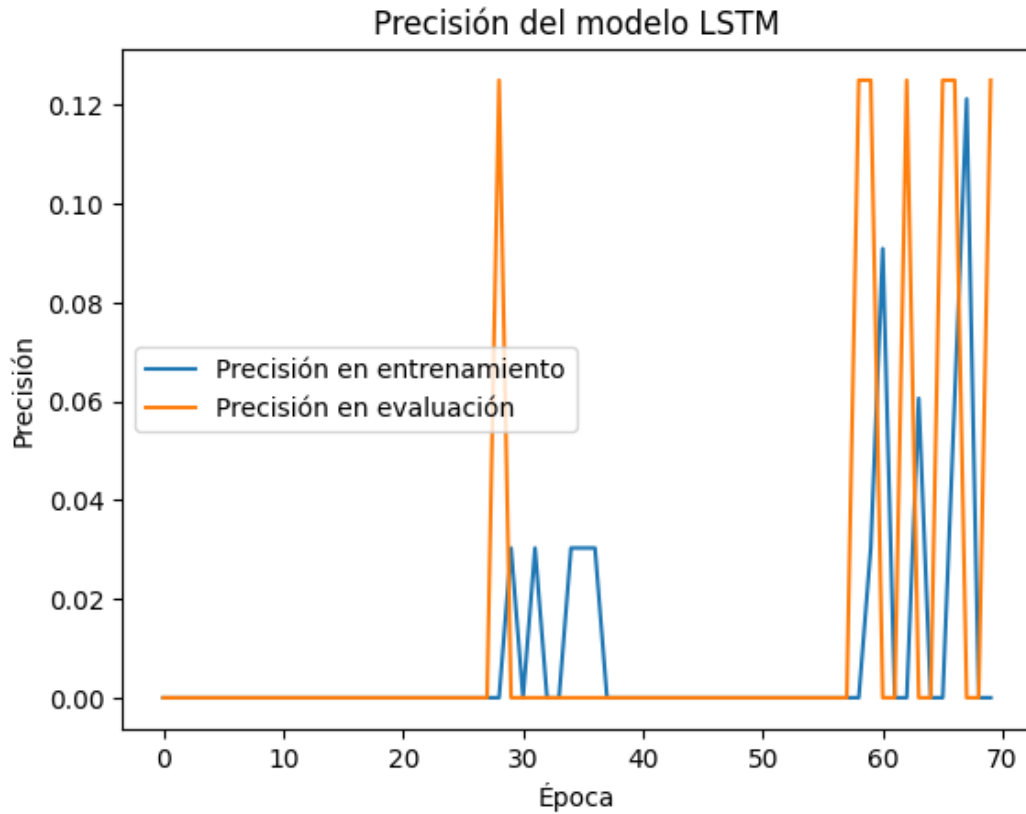


Figura 10

Grafica del accuracy vs epoch



4. Entrenamiento del modelo

Se realiza un entrenamiento del modelo para pacientes personalizados, usando solo los episodios o curvas de glucosa de cada uno de los 10 paciente en cada modelo, y otro modelo generalizado, es decir usando todos los episodios o curvas de glucosa de todos los pacientes en un solo modelo. Con las entradas y salidas que se mencionaron en el capítulo 2.4. Durante el entrenamiento, se ajustan los pesos de la red para minimizar la función de pérdida MSE usada en el modelo.

El mejor modelo es el que tiene el menor error de validación durante el entrenamiento, y es el que se usa para realizar las predicciones con los datos de prueba, o sea el modelo que va a predecir mejor en casos nuevos. Se realiza la validación cruzada K-Fold, usando 5 pliegues, que es una cantidad considerable para que el modelo tenga un tamaño del conjunto de datos con suficientes muestras en cada pliegue para propósitos de error de estimación, esto puede garantizar la fiabilidad del modelo (Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S., 2012).

4.1 Validación cruzada K-Fold

La validación cruzada K-Fold es una técnica comúnmente utilizada en el aprendizaje automático, con el fin de evaluar el rendimiento de un modelo de manera más precisa. La idea detrás de la validación cruzada K-Fold es dividir el conjunto de datos en K "pliegues" o "folds", donde cada pliegue se utiliza una vez como conjunto de prueba y el resto de los pliegues se utilizan como conjunto de entrenamiento (Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S., 2012).

Para la validación cruzada se divide los datos en varios conjuntos, o pliegues de entrenamiento y validación. Estos pliegues se seleccionan aleatoriamente y se utilizan para entrenar el modelo en una serie de iteraciones. En cada iteración, el modelo se entrena en un conjunto de pliegues y se valida en otro conjunto de pliegues. Así que el conjunto de datos con el que se va a realizar el entrenamiento del modelo, se divide en K partes más o menos del mismo tamaño, en este modelo los datos se han dividido en 5 partes y se entrena tantos modelos como partes se haya dividido el conjunto de datos.

Para el primer modelo, se toman los cuatro primeros bloques y se utilizan para entrenar al modelo, y se deja aparte el quinto bloque porque se utiliza para estimar el error de validación.

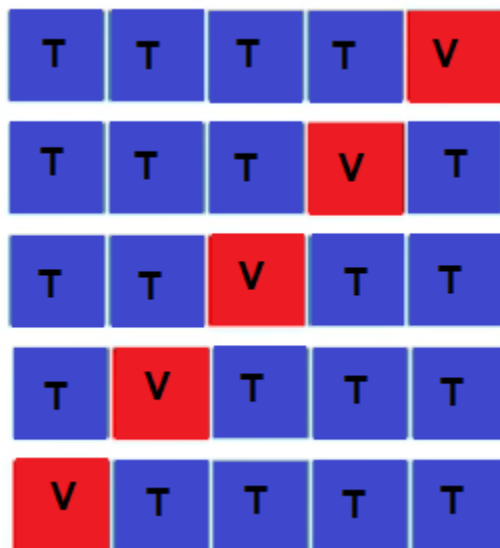
Entonces se ajusta el modelo con los 4 bloques, se hacen las predicciones sobre el quinto bloque que el modelo no ha visto, y se guarda el error.

Ahora se cambia un bloque, y se utilizan 4 bloques para ajustar el segundo modelo, y ese segundo modelo se evalúa en el cuarto bloque, y así se tiene una segunda estimación sobre el error de validación, y al final se tiene la media de cada modelo y la desviación estándar. Y se sigue haciendo K veces ese mismo procedimiento.

La Figura 11 representa las 5 partes en las que se divide el conjunto de mediciones, para realizar la validación cruzada K-Fold, en la que cada modelo usa cuatro partes para el entrenamiento, que son los bloques en color azul, y la quinta parte que son los bloques en color rojo, que se usan para el error de validación en cada modelo. Al final el mejor modelo de las 5 iteraciones, es el que se usa para realizar las predicciones con los datos de prueba, que no ha visto el modelo.

Figura 11

Validación cruzada K-Fold con 5 pliegues



5. Evaluación del modelo

Una vez que se ha realizado la validación cruzada K-Fold, el modelo se ha entrenado y evaluado en 5 particiones de los datos, se procede a evaluar el rendimiento del modelo en el conjunto de prueba que no se utilizó en el entrenamiento del modelo.

5.1 Desnormalización

Para evaluar el rendimiento del modelo, los valores de los datos predichos se desnormalizan aplicando la transformación inversa del objeto StandardScaler. Es decir, se pasan a su escala real, para poder evaluar el rendimiento del modelo correctamente usando las métricas de error MAE y RMSE.

5.1 Métrica de error MAE

La métrica de error MAE, que en español significa Error Absoluto Medio, es una medida de evaluación comúnmente utilizada para medir el rendimiento de un modelo de regresión. El MAE se define como la media de las diferencias absolutas entre las predicciones del modelo y los valores reales.

Para calcular el MAE, primero se toman las diferencias absolutas entre las predicciones del modelo y los valores reales. Luego, se calcula la media de estas diferencias, lo que proporciona una medida de la magnitud promedio de los errores de predicción en términos de la escala de los datos.

La fórmula de la métrica de error MAE (Mean Absolute Error) es la siguiente:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

donde:

- n es el número de observaciones en el conjunto de datos

- y_i es el valor real de la i -ésima observación
- \hat{y}_i es el valor pronosticado por el modelo para la i -ésima observación

El MAE se expresa en las mismas unidades que los datos de entrada, lo que lo hace fácil de interpretar. Un valor de MAE más bajo indica que el modelo tiene una mejor capacidad para predecir los valores reales, mientras que un valor de MAE más alto indica que el modelo tiene una mayor dispersión en las predicciones y, por lo tanto, es menos preciso.

5.2 Métrica de error RMSE

La métrica de error RMSE (Root Mean Squared Error), que en español significa La Raíz del Error Cuadrático Medio, es una medida de evaluación comúnmente utilizada para medir el rendimiento de un modelo de regresión. El RMSE se define como la raíz cuadrada de la media de las diferencias cuadráticas entre las predicciones del modelo y los valores reales.

Para calcular el RMSE, primero se toman las diferencias cuadráticas entre las predicciones del modelo y los valores reales. Luego, se calcula la media de estas diferencias cuadráticas y se toma la raíz cuadrada del resultado, lo que proporciona una medida de la magnitud promedio de los errores de predicción en términos de la escala de los datos.

La fórmula de la métrica de error RMSE (Root Mean Squared Error) es la siguiente:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4)$$

donde:

- n es el número de observaciones en el conjunto de datos
- y_i es el valor real de la i -ésima observación
- \hat{y}_i es el valor pronosticado por el modelo para la i -ésima observación

El RMSE se expresa en las mismas unidades que los datos de entrada, lo que lo hace fácil de interpretar. Un valor de RMSE más bajo indica que el modelo tiene una mejor capacidad para predecir los valores reales, mientras que un valor de RMSE más alto indica que el modelo tiene una mayor dispersión en las predicciones y, por lo tanto, es menos preciso.

6. Resultados Obtenidos

Tabla 1

Resultados de las métricas de error de los 10 modelos personalizados, y el modelo generalizado

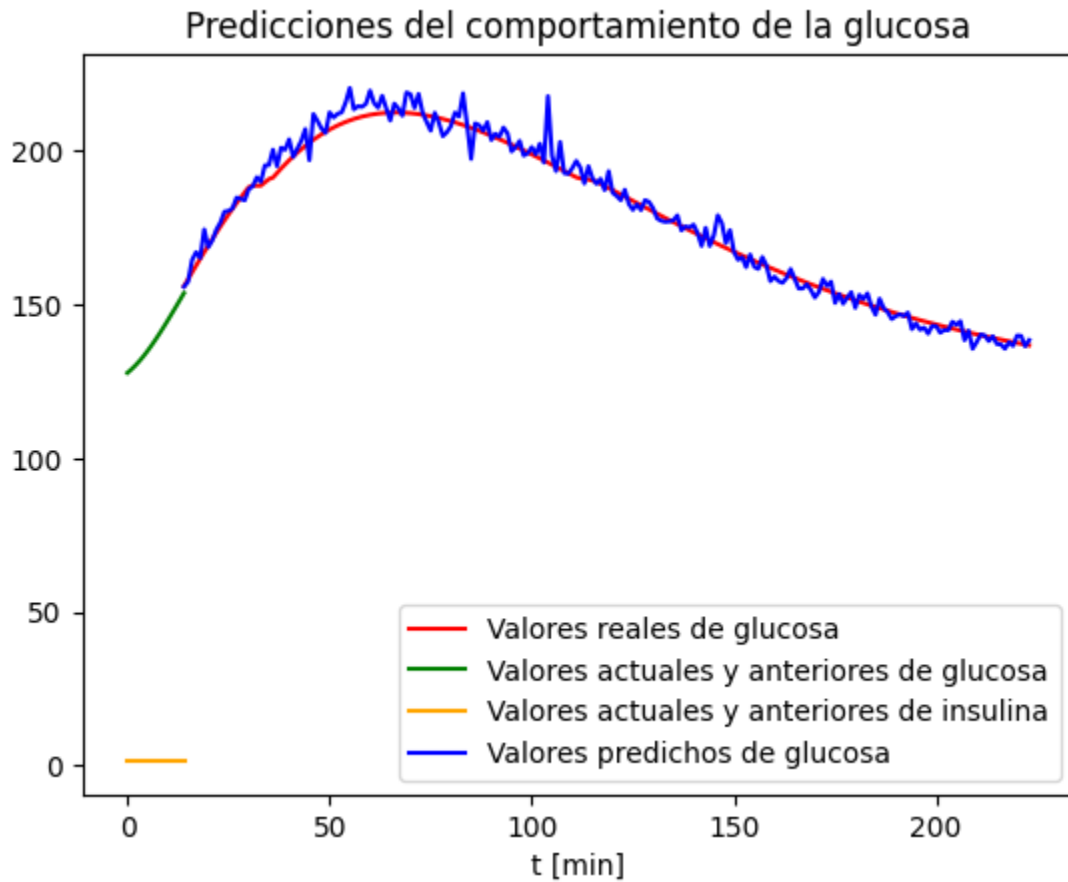
Pacientes	RMSE	MAE	Media MSE	Desviación estándar MSE
Adulto 01	12.0184	9.0252	144.4437	12.0184
Adulto 02	3.6274	2.6148	13.1585	3.6274
Adulto 03	13.6543	9.4404	186.4414	13.6543
Adulto 04	10.4189	7.8268	108.5537	10.4189
Adulto 05	14.6356	10.0531	214.2021	14.6356
Adulto 06	31.6877	23.7926	1004.1135	31.6877
Adulto 07	62.3038	49.8597	3881.7748	62.3038
Adulto 08	36.3444	29.9782	1320.9209	36.3444
Adulto 09	11.5966	8.4199	134.4827	11.5966
Adulto 10	9.4083	6.3796	88.5164	9.4083
Modelo generalizado	20.4676	16.8257	418.9247	20.4676

Las Tabla 1 presenta los resultados al calcular los errores en las predicciones de los modelos personalizados en cada uno de los 10 pacientes adultos, y del modelo generalizado,

usando la escala real de las predicciones hechas a los datos de prueba, realizadas por el mejor modelo durante la validación cruzada K-Fold.

Figura 12

Gráfica de la predicción de la glucosa para el modelo personalizado del paciente adulto 02

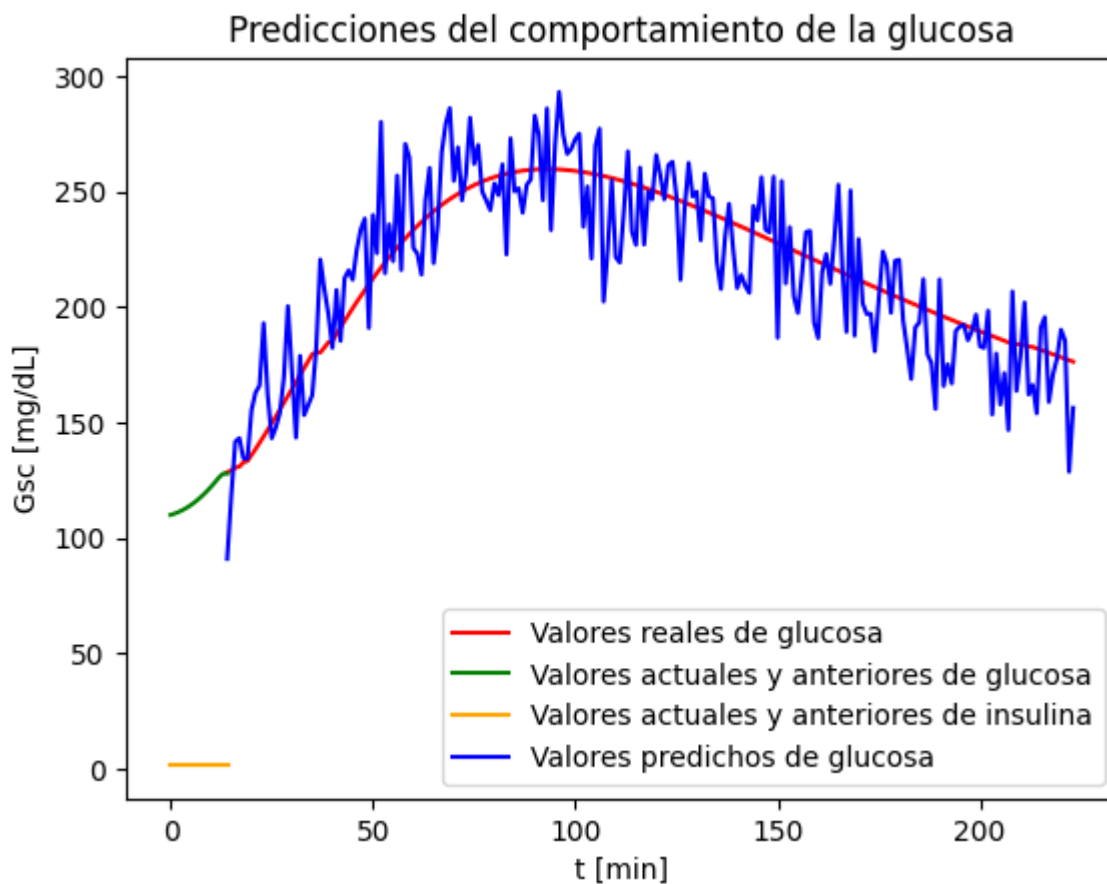


En la Figura 12 se pueden visualizar los resultados del modelo que tuvo el mejor resultado en las métricas de error MAE y RMSE, que fue el modelo personalizado del paciente adulto 02. Para comprender mejor la capacidad predictiva del modelo, en esta gráfica se hace la comparación de los valores reales de los niveles de glucosa en rojo, con las predicciones del modelo en azul, en

un gráfico de líneas. También se muestra en verde y amarillo los datos de prueba de glucosa medida e insulina, con los que se realizaron las predicciones del modelo.

Figura 13

Gráfica de la predicción de la glucosa para el modelo generalizado



En la Figura 13 se pueden visualizar los resultados del modelo generalizado, que fue entrenado con los episodios de todos los pacientes adultos. Para comprender mejor la capacidad predictiva del modelo generalizado, en esta gráfica se hace la comparación de los valores reales de los niveles de glucosa en rojo, con las predicciones del modelo en azul, en un gráfico de líneas. También se muestra en verde y amarillo los datos de prueba de glucosa medida e insulina, con los que se realizaron las predicciones del modelo.

7. Análisis de resultados

Se puede ver la capacidad del modelo, para generalizar bien a nuevos datos mediante la evaluación de su rendimiento en el conjunto de prueba, con las métricas de error RMSE y MAE que se encuentran en la Tabla 1. Es importante tener en cuenta que cuanto menor sea el valor de RMSE y MAE, mejor será el rendimiento del modelo, ya que indica que las predicciones del modelo están más cerca de los valores reales.

En este caso, para el paciente adulto 02 como se muestra en la Tabla 1, el valor de MAE de 2.6148 indica que, en promedio, las predicciones del modelo tienen una diferencia absoluta de 2.6148 miligramos por decilitro (mg/dL) con respecto a los valores reales de glucosa en sangre. Y también para el mismo paciente adulto 02 el valor de RMSE de 3.6274, que indica que la raíz cuadrada del error cuadrático medio es de 3.6274 miligramos por decilitro (mg/dL).

También se puede saber si el modelo está generalizando bien, o si hay overfitting, observando si la desviación estándar MSE es mayor que la media MSE de los datos predichos, o si el modelo es inestable, y como se puede ver en la Tabla 1, la desviación estándar MSE siempre es menor que la media MSE, esto quiere decir que el modelo es estable, y está generalizando bien los datos de prueba.

En la Figura 12 se puede observar que el número de muestras de los datos de prueba de glucosa e insulina en amarillo y verde, que se utilizan para hacer las predicciones de glucosa en azul, son suficientes para predecir el episodio o curva de glucosa que se encuentra en rojo, con valores aceptables en las métricas de error para los datos predichos, ya que las predicciones del modelo siguen de cerca los valores reales de los niveles de glucosa.

Si se usara una ventana de tiempo mayor en la secuencia de entrada, se podría tener una mayor precisión en los valores predichos, pero la idea es tener un tiempo corto, en este caso de 15 minutos, para poder hacer las estimaciones de ingestas ingeridas lo más rápido posible, ya que si uso más tiempo en la secuencia de entrada, el páncreas artificial no podrá seguir el perfil de glucosa de un paciente sano oportunamente.

Y las 3 horas y 30 minutos en la secuencia de salida, influyen también en la precisión del modelo, es decir entre más tiempo tenga que predecir el modelo mayor será el error, pero es el tiempo justo para cubrir la curva de glucosa, que es lo que se quiere predecir para poder hacer la estimación de la ingesta ingerida por el paciente.

8. Conclusiones

Se pudo predecir el comportamiento de los niveles de glucosa en sangre para cada uno de los modelos personalizados de los 10 pacientes adultos con T1DM, y del modelo generalizado de todos los pacientes, a partir de modelos de RNN tipo LSTM, con resultados favorables de las métricas de error MAE y RMSE, que muestran un buen desempeño del modelo para realizar las predicciones.

Los valores de RMSE al ser mayores que los de MAE, indican una mayor dispersión en los errores de predicción, sin embargo, al encontrarse en un rango más amplio de valores de glucosa, las métricas de error pueden considerarse aceptablemente bajas.

La arquitectura definida para el modelo con una capa de entrada, una capa oculta LSTM de 60 celdas, y 210 neuronas en la capa densa, entregaron un equilibrio para que el modelo realizará el entrenamiento de los episodios de glucosa, evitando el sobre ajuste, y permitiendo que las predicciones fueran las más precisas posibles.

Se realizó el entrenamiento usando la técnica de validación cruzada K-Fold, evaluando el rendimiento del modelo, con el monitoreo de la función de pérdida MSE, y el error de validación a lo largo del tiempo, utilizado para medir la diferencia entre la salida predicha por el modelo y la salida verdadera, con un buen rendimiento en la tarea de regresión, es decir, el modelo está haciendo predicciones precisas y cercanas a los valores reales, permitiendo guardar el mejor modelo que realiza la predicción con nuevos conjuntos de mediciones o datos de prueba no vistos durante el entrenamiento.

Se logra para el modelo personalizado del paciente adulto 02 el mejor desempeño con métricas de error RMSE de 3.6274 mg/dL y MAE de 2.6148 mg/dL, para un horizonte de

predicción de glucosa de 3 horas y 30 minutos, con 15 minutos de antelación, que permiten hacer una estimación aproximada de las ingestas ingeridas por el paciente, y de esta manera ayudar a que el páncreas artificial pueda seguir el comportamiento de la glucosa de una persona sana.

La precisión en el modelo personalizado resulta en muchos casos ser más efectiva que en el modelo generalizado, ya que puede capturar la variabilidad individual debido a que la diabetes es una enfermedad compleja que afecta a cada persona de manera diferente, así que utiliza datos específicos de la condición de cada individuo, también tiene una mejor adaptabilidad y mejor capacidad de respuesta a cambios, esto conduce a predicciones más precisas.

9. Recomendaciones

Este proyecto podría verse como un componente, si lo vemos como una parte de un proyecto mayor que busca la implementación de una aplicación móvil que permita llevar el registro del glucómetro, y la insulina que se inyecta la persona, capaz de predecir en pacientes diabéticos, los niveles de glucosa en sangre en tiempo real.

Referencias Bibliográficas

- Abdatum. (2023). Redes LSTM [Captura de pantalla]. Recuperado de <https://abdatum.com/tecnologia/redes-neuronales-recurrentes>
- Alammar, J. (2021). Natural Language Processing with Sequence Models. Coursera. Recuperado de <https://www.coursera.org/learn/nlp-sequence-models/home/week/1>
- Análisis y Programación OOP. (2020). Introducción a las redes neuronales [Captura de pantalla del blog]. Recuperado de <https://analisisyprogramacionoop.blogspot.com/2020/05/introduccion-redes-neuronales.html>
- Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012, April). The 'K' in K-fold Cross Validation. In ESANN (pp. 441-446).
- Celik, M. G., & Varli, S. (2022). Deep Learning Approaches for Type-1 Diabetes: Blood Glucose Prediction. En 2022 7th International Conference on Computer Science and Engineering (UBMK) (pp. 1-5). Diyarbakir, Turkey. <https://doi.org/10.1109/UBMK55850.2022.9919446>.
- Codificando bits. (2019, julio 20). ¿Qué es una red LSTM?. [Video en línea]. Codificando bits. Recuperado de <https://www.youtube.com/watch?v=1BubAvTVBYs&t=388s>
- De Bois, M., Yacoubi, M. A. E., & Ammi, M. (2019). Prediction-Coherent LSTM-Based Recurrent Neural Network for Safer Glucose Predictions in Diabetic People. En T. Gedeon, K. Wong, y M. Lee (Eds.), Neural Information Processing. ICONIP 2019 (vol. 11955, pp.). Lecture Notes in Computer Science. Springer, Cham. doi: 10.1007/978-3-030-36718-3_43.

- De Faria, R. B. F. (2022). Finding patterns that predict hyper and hypoglycemia.
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn & TensorFlow. O'Reilly Media.
- Iacono, F., Magni, L., & Toffanin, C. (2022). Personalized LSTM models for glucose prediction in Type 1 diabetes subjects. En 2022 30th Mediterranean Conference on Control and Automation (MED) (pp. 324-329). Vouliagmeni, Greece. <https://doi.org/10.1109/MED54222.2022.9837153>.
- Idriss, T. E., Idri, A., Abnane, I., & Bakkoury, Z. (2019). Predicting Blood Glucose using an LSTM Neural Network. En 2019 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 35-41). Leipzig, Germany. doi: 10.15439/2019F159.
- Katsarou, A., Gudbjörnsdóttir, S., Rawshani, A., et al. (2017). Type 1 diabetes mellitus. Nature Reviews Disease Primers, 3, 17016. <https://doi.org/10.1038/nrdp.2017.16>
- Katsarou, D. N., Georga, E. I., Christou, M., Tigas, S., Papaloukas, C., & Fotiadis, D. I. (2022). Short Term Glucose Prediction in Patients with Type 1 Diabetes Mellitus. En 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (pp. 329-332). Glasgow, Scotland, United Kingdom. <https://doi.org/10.1109/EMBC48229.2022.9870889>.
- Liang, H., Zhang, S., Sun, J., He, X., Huang, W., Zhuang, K., & Li, Z. (2019). Darts+: Improved differentiable architecture search with early stopping. arXiv preprint arXiv:1909.06035.
- Ng, A. (2018). Neural Networks and Deep Learning. Coursera. Recuperado de <https://www.coursera.org/learn/neural-networks-deep-learning/home/info>
- OMS. (2022). Diabetes. Organización Mundial De La Salud. <https://www.who.int/es/news-room/fact-sheets/detail/diabetes>

- Shuvo, M. M. H., & Islam, S. K. (2023). Deep Multitask Learning by Stacked Long Short-Term Memory for Predicting Personalized Blood Glucose Concentration. *IEEE Journal of Biomedical and Health Informatics*, 27(3), 1612-1623. <https://doi.org/10.1109/JBHI.2022.3233486>.
- Visentin, R., Campos-Náñez, E., Schiavon, M., Lv, D., Vettoretti, M., Breton, M., Kovatchev, B. P., Dalla Man, C., & Cobelli, C. (2018). The UVA/Padova Type 1 Diabetes Simulator Goes From Single Meal to Single Day. *Journal of Diabetes Science and Technology*, 12(2), 273-281. <https://doi.org/10.1177/1932296818757747>
- Zou, F., Shen, L., Jie, Z., Zhang, W., & Liu, W. (2019). A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition* (pp. 11127-11135).

Apéndices

Apéndice A. Link al repositorio con el código de python en google colab, los datos y las pruebas personalizadas y generales de los 10 pacientes.

https://github.com/ismaelVR10/Pruebas_Modelos.git