Alumno: Paulo Gaston Meira Strazzolini

Profesores: Josefina Bompenseri, Tomas Prudente

Materia: Matemática III

Predicción de Diabetes con Redes Neuronales

La diabetes es una enfermedad crónica caracterizada por la hiperglucemia, el aumento de los niveles de glucosa en sangre. Puede ser una enfermedad de nacimiento o desarrollarse a lo largo de la vida del individuo por factores como una alimentación poco saludable, inactividad física, obesidad y la edad. Esta enfermedad puede provocar complicaciones a largo plazo que pueden afectar varios órganos y sistemas del cuerpo, incluyendo enfermedades cardiovasculares, daño a los nervios y problemas en los riñones. Actualmente, se estima que la diabetes afecta al 10% de la población mundial, convirtiéndola en un problema de salud pública significativo. En este informe, exploraremos la posibilidad de utilizar redes neuronales para apoyar al diagnóstico de esta enfermedad.

1. ANÁLISIS DE LA BASE DE DATOS

El set de datos utilizado es una colección de datos médicos y demográficos de personas junto a su diagnóstico de diabetes.

1.1 *Descripción de las columnas*

- **gender**: Género. Es una variable categórica. Hay dos categorías, male = hombre y female = mujer.
- **age**: Edad. Tiene valores en un rango de 0 a 80. Es una variable continua.
- **hypertension**: Hipertension, condición médica en la que la presión sanguínea en las arterias del paciente se halla constantemente elevada. Tiene valores de 0 o 1, donde 0 indica que no tienen hipertensión y 1 indica que tienen hipertensión. Es una variable discreta.

- **heart_disease**: Enfermedad cardíaca, condición médica asociada con un riesgo elevado de desarrollar diabetes. Tiene valores de 0 o 1, donde 0 indica que no tienen enfermedades cardiacas y 1 indica que tienen enfermedades cardiacas. Es una variable discreta.
- **smoking_history**: Historial de tabaquismo, se considera un factor de riesgo para la diabetes y puede exacerbar las complicaciones asociadas a la misma. Es una variable categórica, las categorías son current = actualmente, former = ex fumador, no info = sin información, not current = no actualmente, never = nunca, ever = de vez en cuando.
- **bmi**: Índice de masa corporal (Body Mass Index), medida de grasa corporal basado en peso y altura. Los valores altos de bmi están asociados con diabetes. Es una variable continua que en nuestro dataset abarca el rango de 10.16 hasta 71.55. Un bmi menor a 18.5 se considera bajo de peso, 18.5-24.9 es normal y 25-29.9 se considera sobrepeso, cualquiera valor mayor a 30 es obeso.
- **HbA1c_levels**: Niveles de hemoglobina A1c (hemoglobina glucosilada), promedio del nivel de azúcar en sangre de los últimos 2-3 meses, un alto nivel de hemoglobina A1c indican un mayor riesgo de desarrollar diabetes. Es una variable continua que abarca el rango de 3.5 a 9.0 en nuestro dataset.
- **blood_glucose_levels**: Niveles de glucosa en sangre en un determinado momento. Niveles altos de glucosa en sangre están asociados con diabetes. Es una variable discreta que abarca el rango de 80 a 300.
- **diabetes**: Es la variable que se intenta predecir. Variable discreta en donde 0 indica ausencia de diabetes y 1 presencia de diabetes.

1.2 *Análisis de Correlación*

Analizaremos la correlación de las variables del set de datos con la columna de diabetes. El coeficiente de correlación de Pearson es una medida de qué tan relacionadas están las variables en un rango de -1 a 1, donde valores cercanos a 0 indican poca o nula correlación, valores cercanos a 1 indican una alta correlación y valores cercanos a -1 indican una alta correlación negativa.

Para analizar la correlación de las variables categóricas (gender, smoking_history) mapeamos las categorías con valores numéricos y los reemplazamos en el set de datos.

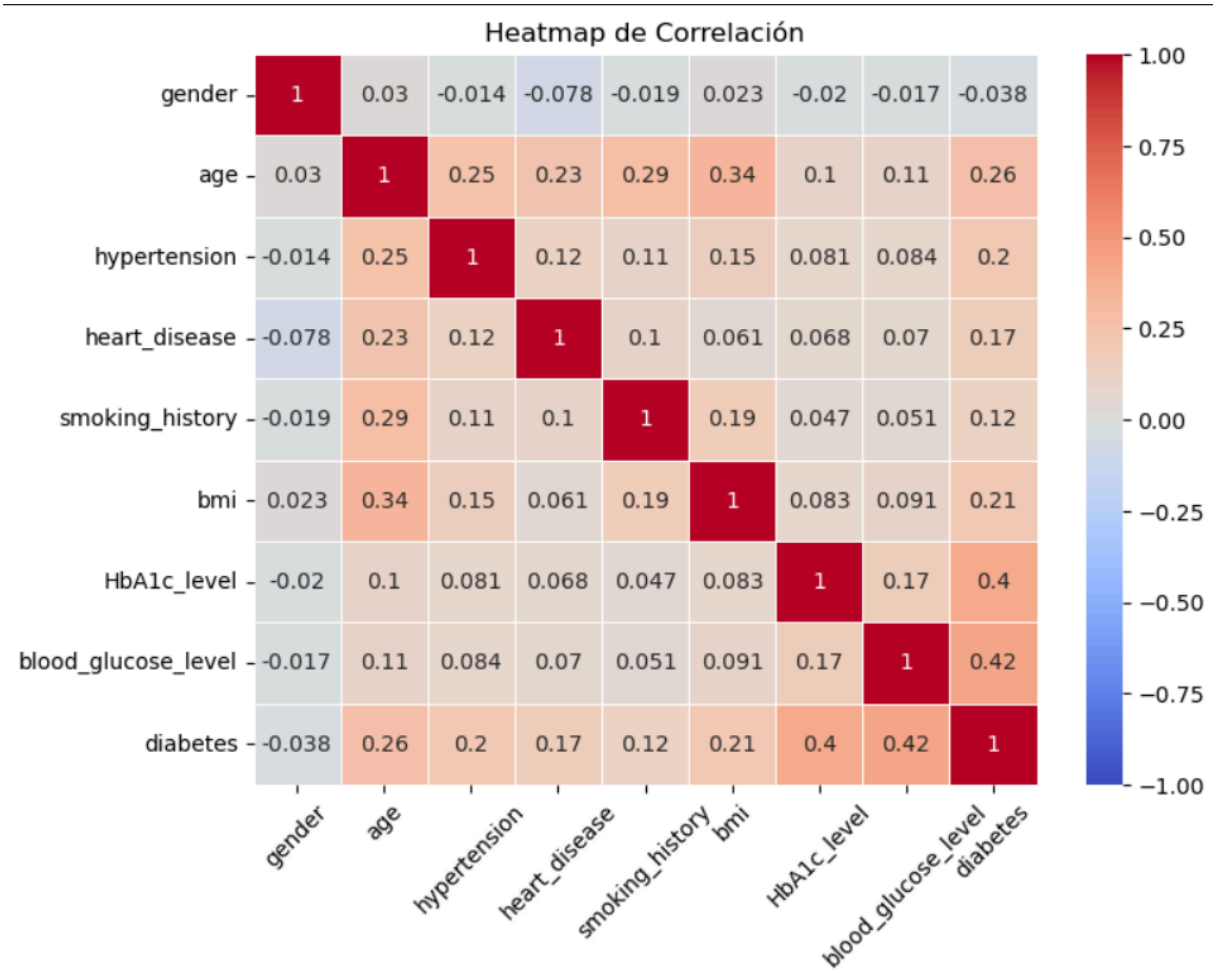

Figura 1
Mapa de calor de los coeficientes de correlación para cada columna del data set

Como se observa en la Figura 1, todas las características tienen un grado de relación con la columna de diabetes y podrían ser tenidas en cuenta para la red neuronal. HbA1c_level y blood_glucose_level son las características con mayor correlación lo cual tiene sentido dado que son fuertes indicadores de la presencia de diabetes.

1.3 Análisis de Factibilidad

El set de datos posee 99982 entradas (filas) por lo que la cantidad de datos se considera suficiente para el entrenamiento de la red neuronal, además de ello los datos presentan

características relevantes para resolver el problema de clasificación y una buena correlación con nuestra variable objetivo que es diabetes. Se observa la predominancia de entradas (filas) con ausencia de diabetes (90% negativo - 10% positivo), pero es esperable para una muestra general de la población.

El propósito del entrenamiento de la red neuronal es que pueda identificar los patrones en los datos y hacer predicciones sobre ejemplos nuevos no vistos durante el entrenamiento. El objetivo final es que pueda proporcionar soporte a la hora de determinar la presencia de diabetes en un individuo.

1.4 Datos Atípicos y Limpieza de Datos

Para la identificación de datos atípicos primero analizamos los histogramas de cada variable. Encontramos cantidades inusuales de entradas en dos variables.

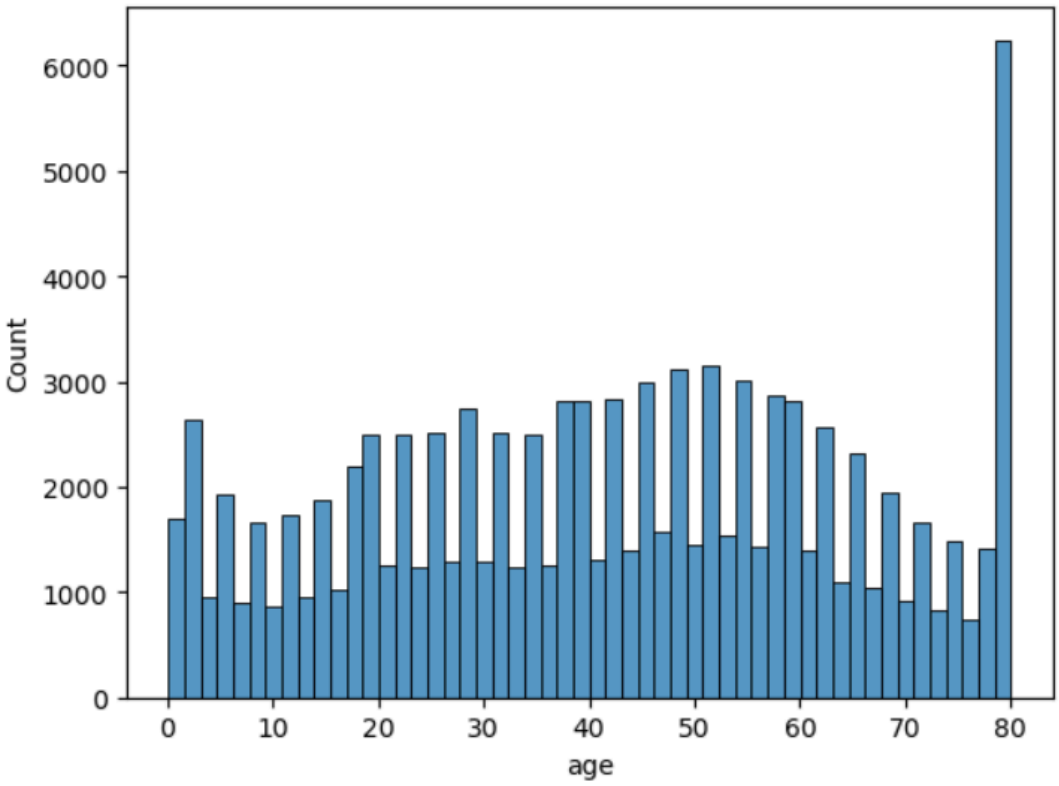

Figura 2
Cantidad de entradas para cada valor de edad

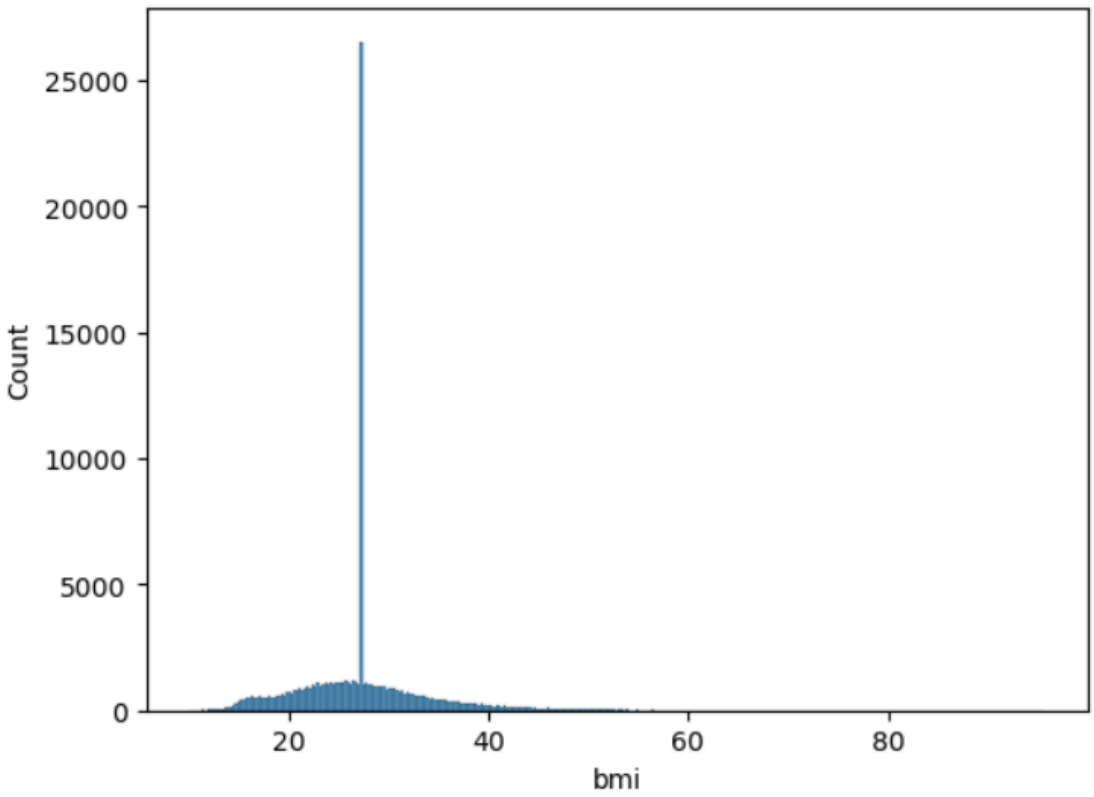

Figura 3
Cantidad de entradas para cada valor de bmi

Tanto en la Figura 2 como en la Figura 3 se observa un sobremuestreo de un particular valor, en el caso de edad el valor es 80 y en el caso del bmi el valor es 27.32. Para el caso del bmi la hipótesis es que los datos fueron preprocesados y cambiaron los valores de null por la media. Considero que estas entradas no son confiables y dada la abundancia de datos en este set me parece oportuno eliminarlas. Para el caso de la edad parecería que la muestra está sesgada y que tuvo un sobremuestreo de personas de 80 años, considero que se debería eliminar ya que no representa correctamente a la población ni tiene coherencia con la distribución que llevaba la muestra. En una población comúnmente se da una distribución normal (en forma de campana), pero también puede tener una distribución hacia la izquierda o derecha dependiendo de la natalidad y la tasa de mortalidad.

Luego de revisados los histogramas de cada variable, analizamos la presencia de datos atípicos en todas las variables no categóricas y con rangos mayores a 0-1 mediante el Rango

Intercuartílico (IQR). El Rango Intercuartílico es una medida de dispersión que indica el rango donde se encuentra la mitad central de los datos, se calcula como la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1), aquellos valores que se hayen a Q1 - 1.5*IQR o Q3 + 1.5*IQR serán considerados atípicos. Se hallaron datos atípicos en las columnas de bmi, HbA1c y blood glucose level.

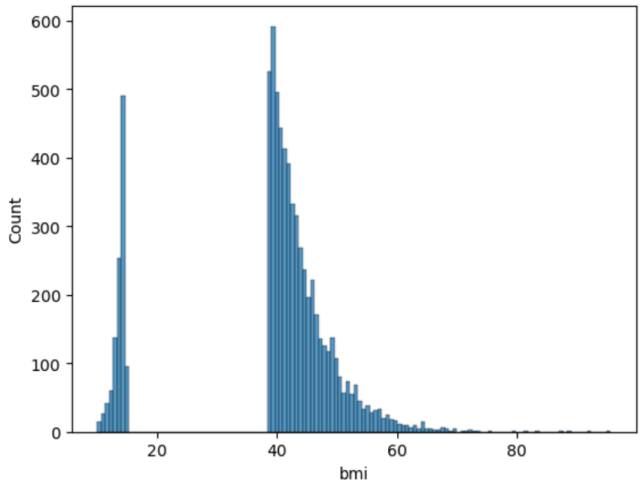

Figura 4
Outliers de bmi

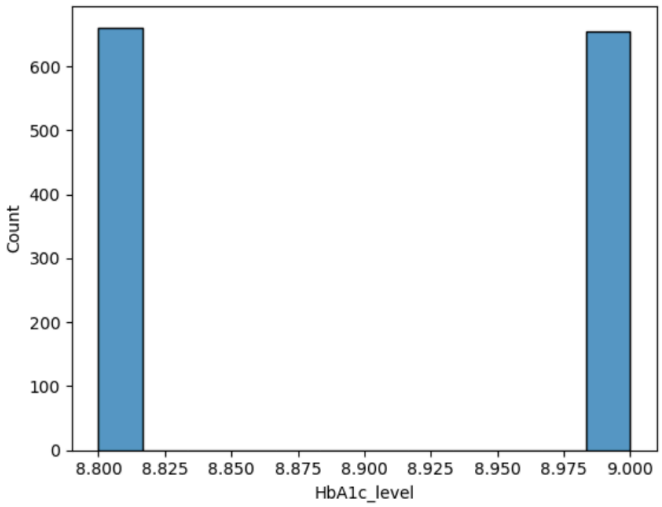

Figura 5
Outliers de HbA1c_level

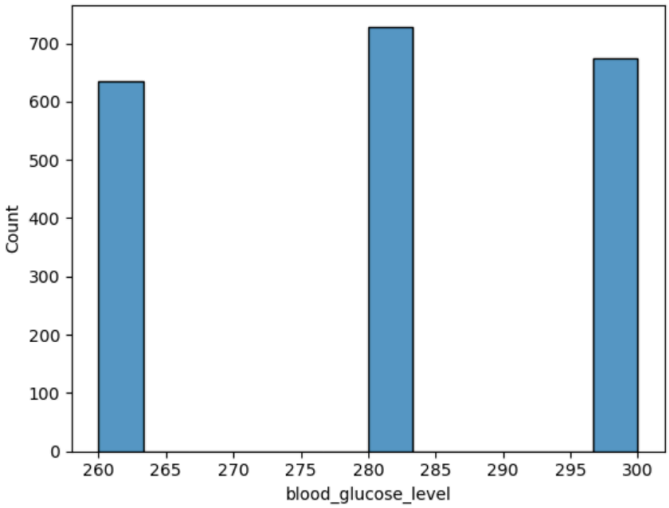

Figura 6
Outliers de blood_glucose_level

Para el caso de bmi eliminamos los valores atípicos directamente. En los casos de HbA1c_level y blood_glucose_level decido dejarlos ya que estos valores son representativos de individuos con diabetes y eliminarlos supondría desbalancear aún más los datos.

Se eliminaron las filas que contenían la categoría “No Info” en la columna smoking_history.

1.5 *Transformaciones Preliminares*

Se transformaron las variables categóricas de gender y smoking_history a variables discretas. Para gender los valores de “male” se mapearon a 0 y “female” a 1. Para smoking_history los valores de “never” se mapearon a 0, los de “current” a 1, los de “not current” a 2, los de “former” a 3 y los de “ever” a 4. Estas transformaciones son necesarias para poder utilizar estos datos en el entrenamiento.

Se normalizaron los datos utilizando Z-score. Con esto logramos que los datos estén en una misma escala y las entradas de la red neuronal no se vean sesgadas por valores grandes.

2. DESARROLLO DE LA RED NEURONAL

2.1 *Arquitectura de la Red*

Para la Red Neuronal utilizaremos una capa oculta con 15 neuronas y una capa de salida con una neurona. Como variables para el entrenamiento elegimos “age”, “hypertension”, “heart_disease”, “smoking_history”, “bmi”, “HbA1c_level” y “blood_glucose_level”. En la capa oculta se emplea como función de activación ReLU y en la capa de salida utilizamos Logistic.

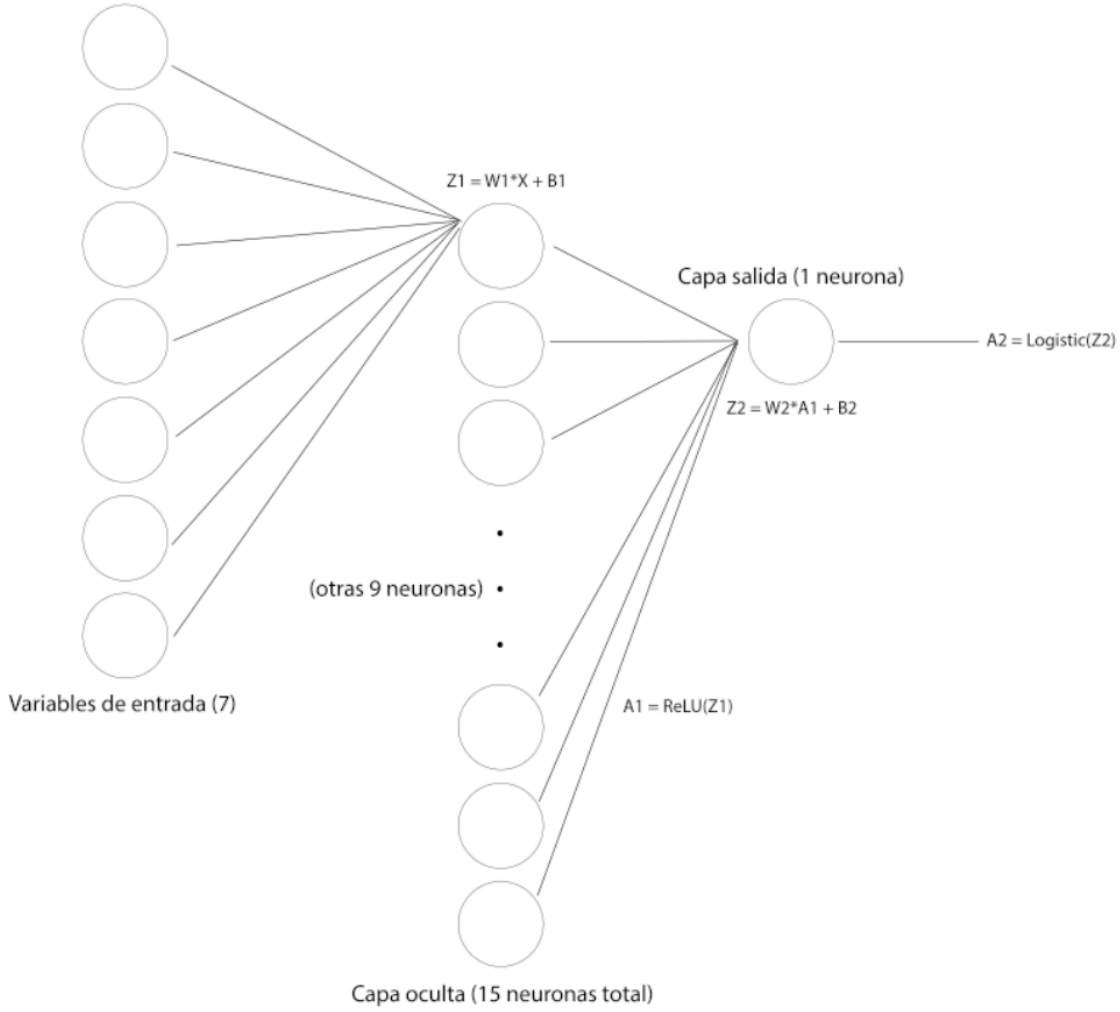

Figura 7
Arquitectura de la Red Neuronal

La función de activación ReLU convierte valores negativos en 0 y deja pasar a los valores positivos sin modificaciones.

$$ReLU(x) = x \text{ si } x>=0 \text{ o } 0 \text{ si } x<0$$

Elegimos esta función para las capas ocultas por su velocidad y capacidad para mitigar el problema del gradiente desvaneciente. La función de activación Logistic convierte los valores a un rango entre 0 y 1.

$$Logistic(x) = 1/(1+e^{-x})$$

Es especialmente útil para problemas de clasificación ya que su salida se puede interpretar como una probabilidad del resultado, si la salida es igual o mayor a 0.5 se interpretará como diabetes positiva, si el resultado es menor a 0.5 se interpretará como diabetes negativa.

2.2 Entrenamiento y evaluación

Para el entrenamiento empleamos el descenso de gradiente con forward propagation y backward propagation. Inicialmente se generan los pesos y sesgos (W y B) con valores pseudo aleatorios, luego durante el descenso de gradiente, en cada iteración, se selecciona una muestra pseudo aleatoria del conjunto de entrenamiento y testeo, se realiza un forward propagation con las variables de entrada.

En cada forward se hace el producto matricial entre la matriz de pesos (W1) por la matriz de variables (X) y se le suma la matriz de sesgos (B1), esto conforma el resultado de la capa oculta (Z1), luego se pasa por la funcion de activacion (ReLU(Z1)) para obtener la salida activada de la capa oculta que luego se utilizara para realizar producto matricial con los pesos de la capa de salida (W2) y se le sumarán los sesgos de la capa de salida (B2), esto conforma el resultado de la capa de salida (Z2) que se pasara por la función de activación (Logistic(Z2)) para obtener la salida activada (A2), que representa la predicción de la red.

En cada iteración, luego del forward propagation se hace un backward propagation, en cada backward se calculan las derivadas parciales para los pesos (W1, W2) y sesgos (B1, B2) de la función de costo, estos valores representa la tasa de cambio en la función de costo y se utilizan para ajustar los pesos y sesgos. Si el resultado de la derivada es positivo, interpretamos que el error sube y reducimos el peso o sesgo, si el resultado es negativo, interpretamos que el error disminuye e incrementamos el peso o sesgo.

En cada iteración del descenso de gradiente se comparan las predicciones de la red con los valores de testeo para la misma muestra, luego se almacenan para realizar los gráficos de precisión y pérdida.

Resultados de la red con 60000 iteraciones.

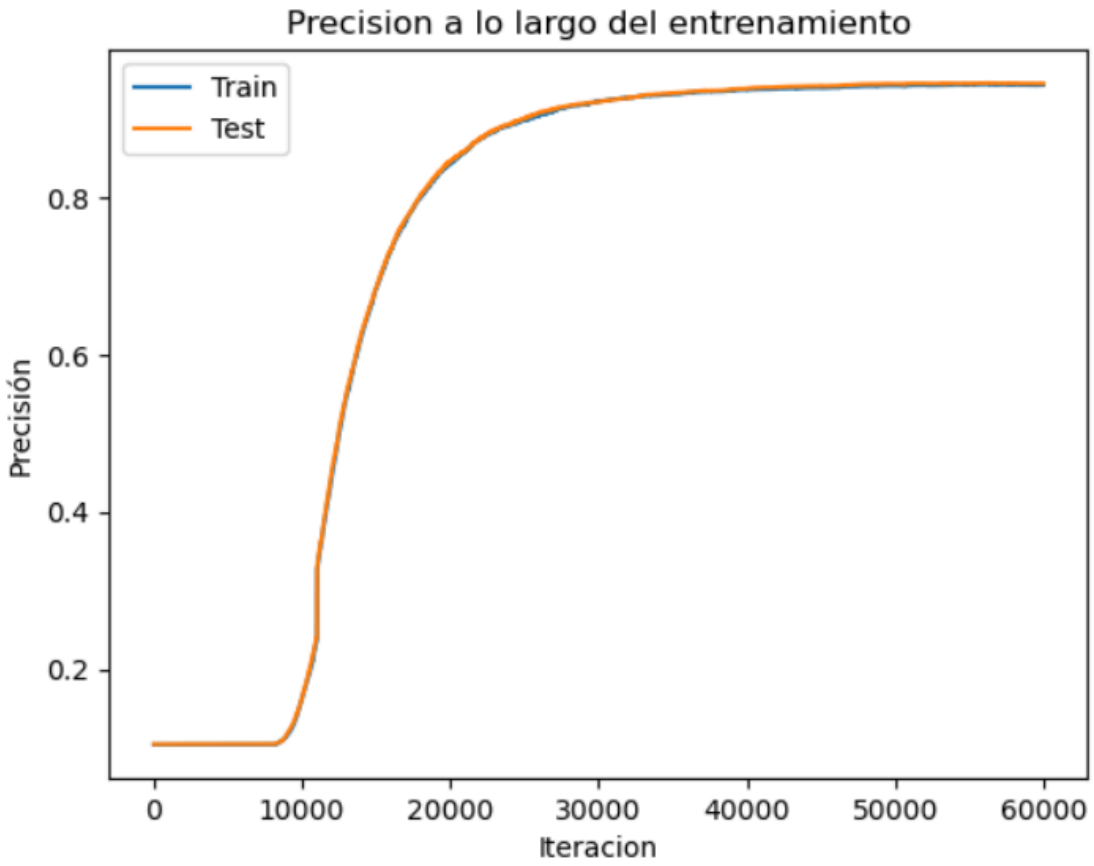

Figura 8

Precisión de la red neuronal para Train y Test por cada Iteración

El resultado del entrenamiento fue una precisión en el conjunto de datos de testeo del 94.29%

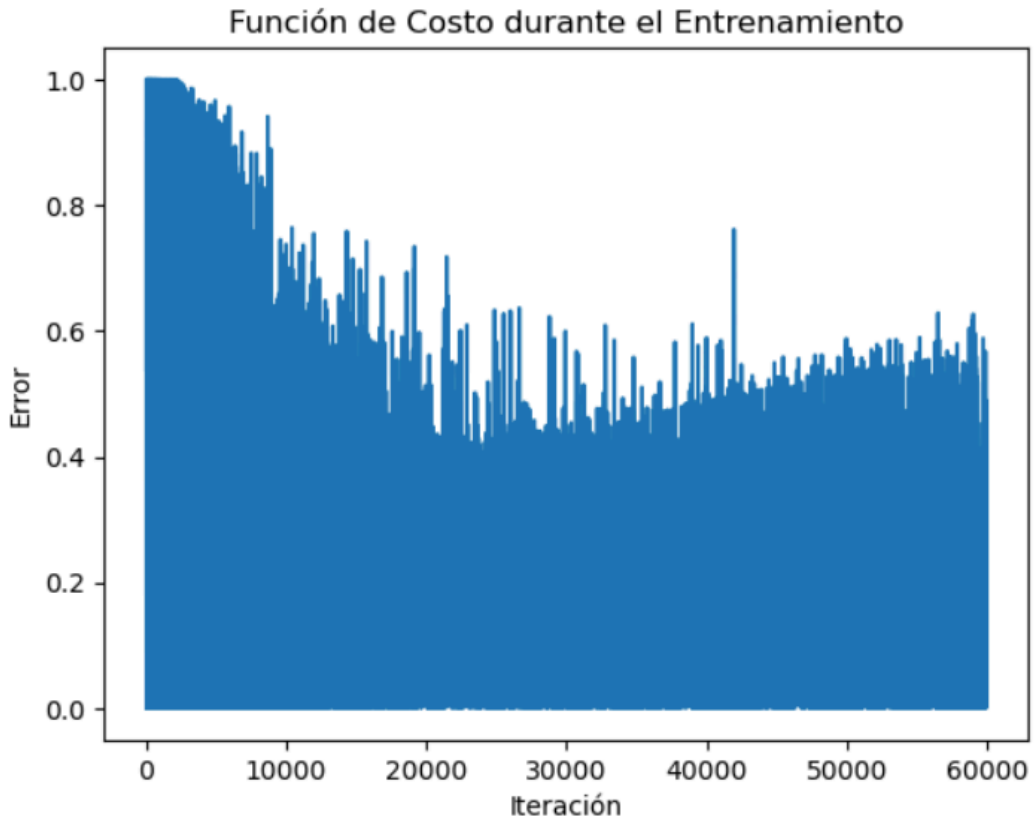

Figura 9
Error cuadrático de la red neuronal para cada iteración

2.3 *Análisis de Overfitting*

Como se observa en la Figura 8 la precisión de las predicciones crece rápidamente hasta aproximadamente las 25000 iteraciones a partir de las cuales se estanca y el aumento de precisión es muy mínimo. A lo largo del entrenamiento se produce muy poca divergencia entre la precisión con los datos de testeo y los de entrenamiento, tampoco se observa que disminuya la precisión de las predicciones en ningún punto del entrenamiento, por lo que es seguro asumir que no se produjo overfitting de la red.

3. COMPARACIÓN CON SCIKIT-LEARN

3.1 Comparación de Rendimiento

Se implementó una red neuronal con la misma configuración que la nuestra en scikit-learn. Como resultado obtuvimos un valor de precisión de 94.76% en un tiempo de entrenamiento mucho menor que en la red neuronal implementada manualmente.

Accuracy: 0.9476121925267572

	precision	recall	f1-score	support
0	0.96	0.99	0.97	14319
1	0.85	0.60	0.70	1658
accuracy			0.95	15977
macro avg	0.90	0.79	0.84	15977
weighted avg	0.94	0.95	0.94	15977

Figura 10
Accuracy_report de sklearn para red con misma configuración a la nuestra

Como se ve en la Figura 8 el accuracy report de sklearn nos provee información adicional sobre el aprendizaje de la red. “precision” mide el porcentaje de predicciones positivas que son correctas, “recall” mide la cantidad de casos positivos detectados contra el total de casos positivos, “f1-score” es una media armónica de “precision” y “recall”. Podemos observar que las estadísticas para la clase de diabetes positiva son menores que las de diabetes negativa, fue posible aumentar la precisión y recall probando otras arquitecturas y configuraciones para la red, como por ejemplo en la Figura 9.

Accuracy: 0.9650368385800402

	precision	recall	f1-score	support
0	0.96	1.00	0.98	13380
1	0.98	0.68	0.80	1550
accuracy			0.97	14930
macro avg	0.97	0.84	0.89	14930
weighted avg	0.97	0.97	0.96	14930

Figura 11
Accuracy_report de sklearn para red con 2 capas ocultas (15 neuronas y 10 neuronas), L=0.005 y 1000 iteraciones

4. CONCLUSIÓN FINAL

Analizamos el set de datos y la correlatividad entre las variables, encontramos los valores atípicos y eliminamos aquellos que consideramos innecesarios, mapeamos las variables categóricas a valores discretos, normalizamos los datos para mitigar las distintas escalas en las que estaba cada variable, dividimos los datos restantes en ⅔ par el entrenamiento y ⅓ para testeo. Creamos la red neuronal con una capa oculta de 15 neuronas, L de 0.0001 y la entrenamos con 60000 iteraciones, empleamos ReLU y Logistic como funciones de activación, El resultado es que obtuvimos una precisión para el conjunto de testeo del 94.28%. Cuando probamos implementar la red con la librería scikit-learn obtuvimos un porcentaje de 94.76%, además empleamos la función de classification_report() de sklearn para obtener información adicional sobre las predicciones de la red neuronal. Mediante estas estadísticas adicionales pudimos concluir que debido al desbalance de clases en la variable diabetes, con mayor presencia de diabetes negativa y poca de diabetes positiva, la predicción de diabetes positiva no es tan precisa como creíamos, ya que la clase mayoritaria es más influye en la métrica de precisión y al no evaluarlas por separado puede pasar desapercibido la verdadera capacidad de la red para predecir la clase minoritaria, por lo tanto considero que la red neuronal en cuestión no debe ser utilizada como soporte real al momento de determinar el diagnóstico de diabetes en un individuo. No descarto la posibilidad de crear una red neuronal más adecuada para la predicción de diabetes, realizando tratamiento adicionales al set de datos (como por ejemplo submuestreo de la clase mayoritaria), probando otro tipo de arquitecturas y configuraciones o utilizando un set de datos completamente diferente.