Universidad Nacional de Córdoba

Facultad de Matemática Astronomía y Física

Predicción de accidentes cardiovasculares a partir de redes neuronales

Trabajo final de Moretti, Ignacio

Dirigido por Dr. Tamarit, Francisco

2014

Resumen

En este trabajo se desarrolló, mediante el uso de redes neuronales artificiales, una herramienta para la predicción de accidentes cardiovasculares como los ataques isquémicos, accidentes cerebrovasculares e infartos de miocardio. La construcción de esta herramienta y su proceso de aprendizaje se baso en un banco de datos de 1686 registros médicos de pacientes canadienses proporcionados por la empresa Lammovil SA. El resultado de este estudio sobre el uso de las redes neuronales artificiales para la predicción de accidentes cardiovasculares culminó con una aplicación para dispositivos móviles con sistema operativo Android. Será utilizada por profesionales del área como una herramienta de ayuda para prevención de dichas afecciones.

Clasificación

- C.1.3 Other Architecture Styles
- D.4.8 Performance
- F.1.1 Models of Computation
- I.2.1 Applications and Expert Systems
- I.2.6 Learning
- I.5.1 Models
- J.3 LIFE AND MEDICAL SCIENCES

Palabras Claves

Redes Neuronales Artificiales; Predicción de accidentes cardiovasculares; Inteligencia Artificial; Aplicaciones Android.

Agradecimientos

A Berta Reisin y Miguel Angel Moretti por el apoyo incondicional a lo largo de toda mi vida y mi carrera. A Lucía Feuillet por el empuje, apuntalamiento, amor y paciencia. A Sebastian Moretti y Cecilia Moretti por sus ejemplos a la distancia y a la vez tan cerca. A Francisco Tamarit y Aldo Algorry por la confianza y la guía en todo el proceso. A Nicolás Wolovick por la ayuda en toda la carrera y corrección final de este trabajo. A Marcelo Feuillet por la bibliografía y ayuda en el campo médico. A Lammovil S.A. a través de Dr. Luis Armando, Dr. Hernán Perez, Dr. Hugo Villafañe y Ing. Aldo Agorry por el banco de datos y la asistencia técnica provista en el campo de la Ingeniería y la Medicina.

Introducción

En este trabajo, nos proponemos predecir accidentes cardiovasculares mediante el uso de redes neuronales.

Las Redes Neuronales Artificiales (RNA) son un modelo computacional inspirado en la estructura de las redes neuronales biológicas. Podemos definirlas como un sistema de aprendizaje que consta de dos fases bien delimitadas, la primera fase llamada de aprendizaje y la segunda de ejecución. En la mayoría de los casos, son sistemas adaptativos, que cambiando su estructura a partir de la información externa e interna fluyente en la red durante la fase de aprendizaje, consiguen dar respuestas adecuadas en la fase de ejecución. Las RNA son capaces de capturar relaciones no lineales entre variables, y si son correctamente organizadas, pueden aproximar cualquier función no lineal. En nuestro caso, intentaremos probar que las RNA pueden encontrar relaciones no lineales entre distintos indicadores clínicos obtenidos de los pacientes. A partir de los datos colectados, lograremos predecir un posible accidente cardiovascular en el transcurso de 5 años posteriores a la evaluación médica.

A este respecto, debemos señalar el importante aporte que esto implica a un problema fundamental de la medicina desde las Ciencias de la Computación. Contar con predictores capaces de orientar al médico en la detección e incluso en el tratamiento de diferentes patologías, puede ser de gran utilidad en el cruce interdisciplinario medicina-computación.

Por su parte, el campo de las redes neuronales, como rama de la inteligencia artificial, ha tenido un gran desarrollo en las últimas décadas. De esta manera, la definición de neuronas artificiales muy simples, conectadas entre sí mediante una compleja arquitectura de conexiones sinápticas y simples reglas dinámicas de actualización de los estados neuronales hace posible desarrollar sistemas neuronales

artificiales capaces de resolver los más variados problemas. Éstos tienen la capacidad, como sucede con los biológicos, de adaptarse al medio y aprender a dar solución a situaciones novedosas.

En el caso de las patologías cardíacas, es casi imposible descubrir de manera manual patrones que predigan accidentes cardiovasculares, dado que la capacidad humana no alcanza a registrar la cantidad de datos necesaria para realizar esta operación. Por esta razón, es necesario utilizar una técnica computacional para analizar estos datos y descubrir de manera automática los patrones que ayudan a predecir dichas afecciones.

A su vez, desarrollaremos una aplicación sobre la plataforma Android, propia a los teléfonos celulares inteligentes o tabletas, que consista en la ejecución de la red neuronal para la predicción de accidentes cardiovasculares. Teniendo en cuenta el auge actual de los artefactos móviles, creemos que será importante utilizar estas nuevas tecnologías con los fines ya mencionados.

Acto seguido, presentaremos el problema a resolver y la fundamentación de su elección para este trabajo. Como mencionamos más arriba, estudiamos aquí la capacidad de las redes neuronales del tipo feedfoward para predecir si un paciente tendrá o no un accidente cardiovascular en los siguientes cinco años. Para esto, disponemos de un conjunto de datos clínicos y de resultados de exámenes médicos realizados a 1689 pacientes que fueron controlados por más de cinco años. Contamos entonces con el dato que queremos predecir en nuevos pacientes: la aparición de un accidente cardiovascular en un tiempo determinado. Nuestro trabajo será entrenar varios tipos de redes neuronales, observar los resultados obtenidos y así estudiar el nivel de precisión de cada una.

Para resolver nuestro problema, nos enfrentamos a distintos inconvenientes: en primer lugar, tenemos en el conjunto de datos más 200 variables a ser tenidas en cuenta; en segundo lugar, no todos los registros están completos, en otras palabras, no poseemos valores en todas las variables; en tercer lugar, los valores de los datos no se encuentran normalizados. Una vez que tenemos el banco de datos listo para ser utilizado, es decir, cuando hemos seleccionadas las variables más importantes en conjunto con un equipo médico, eliminado los registros incompletos y normalizado el conjunto de datos, nos enfrentamos al desafío de construir varias redes neuronales

y entrenarlas, para luego verificar los resultados y documentar todo el proceso. Por último y a modo de innovación, diseñaremos un programa para una plataforma móvil Android que ejecute la red neuronal y así obtendremos una herramienta que no sólo funcione para la predicción de un accidente cardiovascular, sino también para la prevención de este tipo de afecciones de salud.

Índice general

Capítulo 1

Objetivos

En este capítulo enunciaremos los objetivos trazados en nuestra investigación y describiremos el objeto de estudio.

El objetivo general es contribuir desde una rama de las ciencias de la computación a ciertos problemas de la medicina, favoreciendo no sólo un diálogo interdisciplinario, sino también la resolución de interrogantes actuales de otras ciencias. Pretendemos crear una herramienta útil tanto para profesionales de la salud como para la sociedad en su conjunto, además de integrar las nuevas tecnologías -aplicaciones móviles- con la teoría de redes neuronales artificiales y los métodos computacionales.

Nuestros objetivos específicos son, por un lado, construir una herramienta y/o programa computacional basado en Redes Neuronales Artificiales, que pueda predecir accidentes cardiovasculares. Esta predicción deberá fundarse en datos reales y ser desarrollada con un nivel de confianza aceptable; con esto nos referimos a que, en comparación con los métodos utilizados actualmente como el coeficiente de Framingham, obtenga mejores resultados. Dichos resultados serán evaluados con métodos numéricos como la especificidad, la sensibilidad y el valor predictivo de una prueba. Por otro lado, apuntaremos a desarrollar esta aplicación específica para el sistema operativo Android. Para cumplir con estos objetivos utilizamos el marco teórico descripto en el capítulo siguiente.

Trabajamos con un banco de datos de estudios médicos realizados a pacientes Canadienses, creamos varias redes neuronales y documentamos los resultados, luego comparamos estos resultados con un método que actualmente se utiliza en la

medicina -el coeficiente de Framingham-.

Nos basamos en este banco de datos para la construcción y verificación de nuesto objeto, las redes neuronales artificiales.

Capítulo 2

Marco Teórico

2.1. Marco Teórico Computacional

2.1.1. Neuronas Artificiales

Antes de entender el concepto de redes neuronales, es primordial comprender la definición del modelo de las neuronas artificiales. En 1943, Warren McCulloch y Walter Pitts[20], basándose en el comportamiento de las neuronas biológicas, propusieron un modelo simple de una neurona artificial, que definieron como una unidad de umbral binaria. Específicamente el modelo calcula la suma de los pesos de las entradas conectadas a las salidas de otras unidades neuronales, y si la suma de esos pesos supera un número predeterminado, la neurona genera como salida un 1; mientras que de no superar dicho umbral, genera un 0. Podemos observar este comportamiento definido en la Ecuación 2.1.

$$n_i(t+1) = \Theta\left(\sum_j w_{ij} n_j(t) - \mu_i\right) \tag{2.1}$$

Donde Θ se define como muestra la Ecuación 2.2.

$$\Theta(x) = \begin{cases} 1 \quad si \quad x \geq 0 \\ \\ 0 \quad si \quad x < 0 \end{cases} \tag{2.2}$$

Para aclarar las ecuaciones anteriores n_i es 1 o 0, y representa el estado activo o no de la neurona i. Luego, t representa discretamente el tiempo o pasos de procesamiento. $\Theta(x)$ es lo que llamamos función de activación. Diremos pesos a los w_{ij}, que representan la fuerza de la conexión sináptica entre la neurona j y la neurona i. Definimos que la conexión puede ser exitadora o inhibidora si su valor es positivo o negativo respectivamente. Este valor también puede ser 0, en tal caso no existe conexión entre dichas neuronas. Por último, μ_i es el parámetro umbral de la neurona i, dado que cuando la suma de las entradas supera este umbral, la neurona se activa. En resumen, éste sería el modelo presentado por McCulloch y Pitts para definir la unidad básica que compone a las redes neuronales artificiales.

2.1.2. Perceptrones

En 1958[9] Frank Rosenblatt publicó un artículo donde se describió por primera vez el concepto de perceptrons. Desde entonces, diferentes arquitecturas han sido desarrolladas. Frank Rosenblatt y su grupo de investigación, se centraron en el problema de cómo encontrar apropiadamente los pesos w_{ij} para una tarea computacional. Desarrollaron una red definida como **perceptrons**, donde las unidades o neuronas están organizadas en diferentes capas que poseen una conexión **feedforward** entre sí.

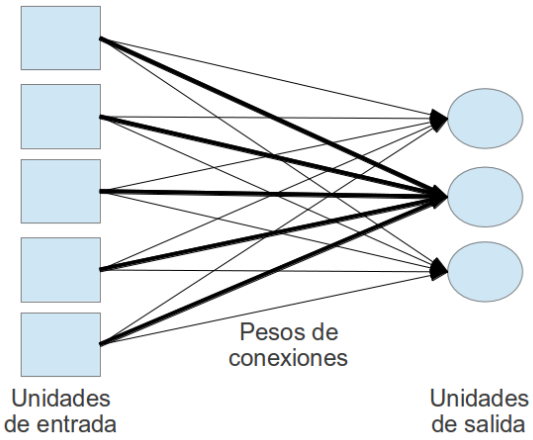

Figura 2.1: Perceptron de 2 capas

En la Figura 2.1 observamos un ejemplo de una red **perceptrons** de dos capas.

Una red similar definida como **adalines**[4] fue desarrollada al mismo tiempo por Bernard Widrow y Ted Hoff. En el año 1962, Rosenblatt pudo probar la convergecia del llamado algoritmo de aprendizaje en una red perceptron sin ninguna capa intermedia[10]. Este algoritmo consistía en ir cambiando los pesos de las conexiones iterativamente. El principal problema de este tipo de redes fue expuesto en la publicación de Marvin Lee Minsky y Seymour Papert[15]. Básicamente el poder expresivo de este tipo de redes es limitado.

Rosenblatt estudió distintas estructuras con más capas e intuía que estas podrían romper las limitaciones de la estructura simple de un perceptron sin capas intermedias. Pero no descubrió ningún algoritmo de entrenamiento que determinara los pesos de las conexiones. Las investigaciones en este aspecto no avanzaron hasta unos 20 años después, cuando varias personas desarrollaron un algoritmo que funcionó correctamente para ajustar los pesos de las conexiones de los perceptrons de múltiples capas. El algoritmo de entrenamiento para este tipo de redes fue conocido como **backpropagation**, y fue publicado por primera vez por de Paul Werbos en 1974[14], y luego por David Rumelhart, Geoffrey Hinton y Ronald J. Williams en 1986[8], pudiéndose así solucionar el problema de expresividad del perceptron de una capa.

2.1.3. Aplicabilidad

Existen varios tipos de problemas que pueden ser resueltos a través del uso de una red neuronal, podemos agruparlos en cuatro tipos básicos: clasificación, predicción, reconocimiento de patrones y optimización.

Clasificación

Es el proceso de distinguir y particionar las entradas de la red neuronal en grupos o categorías. Por ejemplo una compañía de seguros quiere clasificar los diferentes seguros que brinda en categorias de diferentes riesgos, o bien una compañía de servicios de correo electrónico quiere distinguir a los correos entrantes entre dos categorias si son o no correo basura.

Predicción

Es el proceso por el cual, entregando a la red neuronal una serie temporal de datos de entrada, ésta puede predecir los futuros valores. Por ejemplo son utilizadas

para predecir los movimientos del mercado financiero.

Reconocimiento de Patrones

Se podría entender como un problema de clasificación pero más específico al campo de las imágenes, es simplemente la habilidad de reconocer patrones incluso cuando estan distorsionados. Por ejemplo la identificación de objetos a través de imágenes, el reconocimiento de imágenes o reconocimiento de caracteres escritos a mano.

Optimización

Son problemas donde se busca una solución de aproximación. Por ejemplo mejorar circuitos integrados, mejorar el uso de memoria o el problema del viajero donde un vendedor tiene que visitar un grupo de ciudades, y debe hacerlo tomando la ruta mas conveniente reduciendo la cantidad de kilómetros.

2.1.4. Entrenamiento y Validación

Una de las etapas más importantes a la hora de desarrollar una aplicación con redes neuronales es la de entrenamiento de la red. El entrenamiento de una red es el proceso por el cual se asignan los pesos definitivos a las conexiones entre las neuronas. Existen varios algoritmos de entrenamiento, pero en general se empieza asignando números aleatorios a los pesos, luego un proceso de validación examina los resultados, y por último los pesos se actualizan según como haya resultado esa validación. Este proceso se repite hasta que el error de la validación sea menor a un valor preestablecido según el problema a resolver. Estos tipos de entrenamientos se dividen en tres grandes grupos o categorías: supervisados, no supervisados o híbridos.

Supervisado

Consiste en entregar a la red un conjunto de datos de entrada y además, la salida esperada para dicha entrada. El proceso de entranamiento se repite hasta que el resultado obtenido se asemeja al resultado esperado con un nivel de error aceptable y predefinido.

No supervisado

Es similar al anterior sólo que no se le entrega el resultado esperado, es usado en general para clasificar las entradas en diferentes grupos.

Híbridos

Existen varios métodos híbridos que combinan aspectos de los dos anteriores.

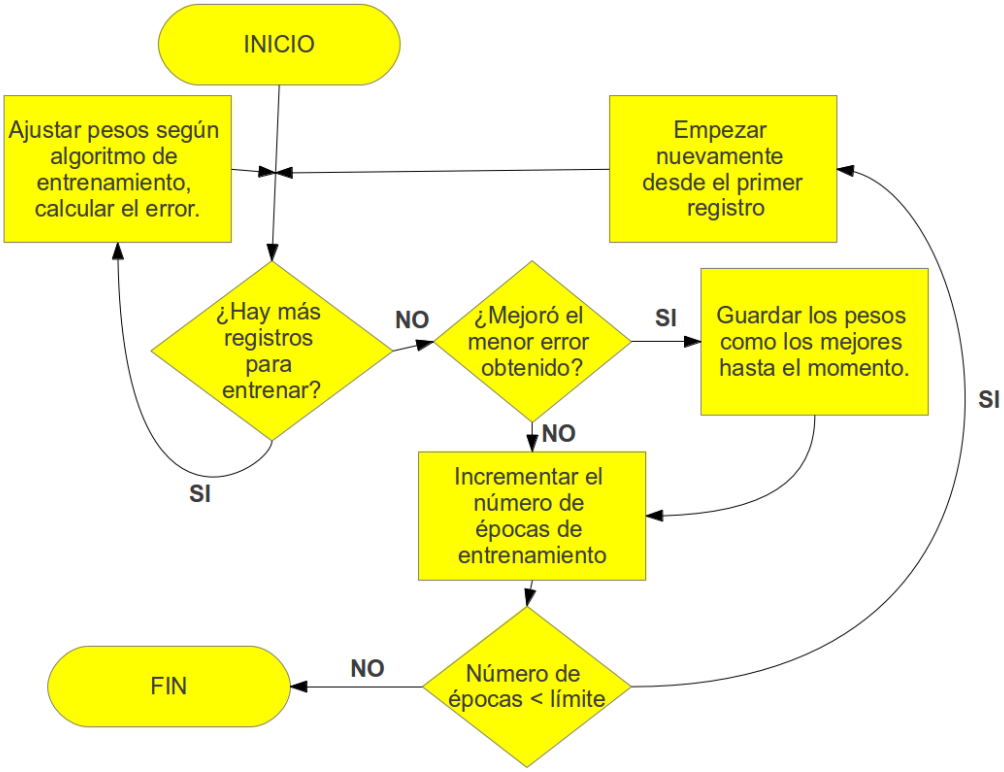

Figura 2.2: Diagrama de flujo de entrenamiento supervisado

En nuestro trabajo, nos centraremos en un algoritmo de entrenamiento supervisado, describimos este tipo de algoritmos con la Figura 2.2 que no es más que un diagrama de flujo sobre el funcionamiento de estos algoritmos en general.
Como dijimos anteriormente, el paso siguiente al entrenamiento es la etapa de validación. Para realizar la validación es necesario que el conjunto de datos utilizado sea distinto al del entrenamiento. Es de suma importancia el correcto mantenimiento de estos datos. El error en la etapa de entrenamiento será generalmente más bajo que el error en la etapa de validación.

2.1.5. Cálculo del Error

Un aspecto importante en el proceso de entrenamiento y validación de las redes neuronales es el cálculo del error. El objetivo de todo algoritmo de entrenamiento es minimizar la tasa de error. Tenemos entonces que considerar dos valores para determinar la tasa de error en algoritmos de entrenamiento supervisado. Primero, debemos calcular el error de cada registro del conjunto de datos de entrenamiento. En segundo lugar, debemos calcular el promedio de todos los errores. El error de salida es simplemente la diferencia entre la salida de la red neuronal con la salida esperada o ideal. Este valor es usado para calcular el error cuadrático medio (RMS), una vez que todos los registros del set de entrenamiento hayan sido procesados por la red podremos calcular el RMS. Este error entonces actúa como una tasa de error global de toda la red neuronal. El error de entrenamiento y el de validación se calculan de la misma manera, utilizando la misma ecuación.

$$error = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(actual_i - ideal_i)^2} \tag{2.3}$$

En la Ecuación 2.3 n representa la cantidad de registros de entrenamiento o de validación, *actual* representa el resultado de la ejecución de la red neuronal para el registro i, *ideal* es el resultado esperado para el registro i. El algoritmo de entrenamiento utiliza este error para modificar los pesos con el objetivo de reducir este error.

2.1.6. Redes Neuronales Feedforward y Backpropagation

El término *feedforward* describe cómo la red procesa los patrones, sus unidades o neuronas sólo están conectadas con las neuronas de la capa siguiente, por lo tanto no existen conexiones con las capas anteriores. *Backpropagation* es una forma de entrenamiento supervisada. Utiliza los resultados esperados y los obtenidos para calcular el error y de este modo ajustar los pesos de las conexiones entre las neuronas desde la salida hacia la entrada de los datos, es por ello que toma el prefijo *back*. Las redes feedforward comienzan con una capa de entrada de patrones o datos, ésta puede estar conectada a una capa oculta o directamente a la capa de salida. Pueden tener

cualquier cantidad de capas ocultas.

Capa de Entrada

Es la interfase de conexión entre el ambiente externo que suministra los datos o patrones y la red neuronal. Cuando se cargan los datos en ella, la capa de salida produce como respuesta un conjunto de datos o patrones de salida. Esto es en esencia lo que hace la red neuronal. Cada neurona de la capa de entrada representa una variable independiente que influye en el resultado de la red neuronal.

Capa oculta

No existe contacto con el ambiente externo. Son necesarias para poder alcanzar el nivel de expresividad que tendrá la red neuronal en función de resolver un problema determinado. El número de capas ocultas normalmente varía entre 1 y 2.

Capa de Salida

Es la capa que interactúa con el ambiente externo otorgando el resultado final de la red neuronal. El número de neuronas es relativo al trabajo que pretende realizar la red neuronal.

Función de Activación

La mayoría de las redes procesan la salida de las distintas capas a través de una función, a la cual llamamos de activación. Esta función normaliza las salidas entre dos valores determinados. La función más utilizada en este tipo de redes en la función sigmoidal, pero también es posible utilizar otras como la tangente hiperbólica, además de crear una función propia diferenciable. Es posible utilizar una función diferente para cada capa. La función Sigmoidal se define en la Ecuación 2.4.

Capas ocultas	Característica
0	Solo pueden representar funciones linealmente separables
1	Pueden aproximar cualquier función continua desde un espacio finito a otro
2	Pueden aproximar cualquier tipo de función

Cuadro 2.1: Caracteristicas según el número de capas ocultas

$$f(x) = \frac{1}{1 + e^{-n}} \tag{2.4}$$

Un aspecto importante de la función sigmoidal es que sólo devuelve valores positivos. En cambio, si necesitamos que la red nos devuelva valores negativos debemos implementar la función de activación tangente hiperbólica. Esta función se define en la Ecuación 2.5.

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{2.5}$$

Números de neuronas y capas ocultas

Otros interrogantes a resolver son cuál es la cantidad de capas ocultas y cuántas neuronas por capas tenemos que emplear en la estructura de una red neuronal. Existen muy pocos problemas que requieran más de dos capas ocultas. Sin embargo, las redes con dos capas ocultas pueden representar funciones de cualquier tipo. No existe alguna razón teórica hasta hoy para utilizar una red neuronal con más de dos capas ocultas, en 1989[11] George Cybenko demostró que con solo una capa oculta es suficiente para aproximar cualquier función continua. Además, para la mayoría de los problemas comunes que resulven las redes neuronales, no hay ninguna razón para usar más de una capa oculta. En el cuadro 2.1 detallamos las características de las distintas estructuras de redes dependiendo del número de capas ocultas.

Pero ahora debemos saber cuál es la cantidad de neuronas por capas que debemos utilizar, si empleamos muy pocas neuronas en las capas ocultas es probable caer en un error llamado **underfitting**. Este problema sucede cuando no podemos identificar los patrones en un set de datos muy complejo. Por otro lado, si utilizamos

demasiadas neuronas por capas nos enfrentamos a varios problemas: el primero es el **overfitting**, que aparece cuando la red tiene mucha capacidad de procesamiento y no hay suficientes datos para entrenar a todas las neuronas en la capa oculta, el segundo, ya teniendo los datos para el entrenamiento, sucede al encontrarnos con demasiadas neuronas, porque el tiempo para entrenar la red crece exponencialmente. Existen varias reglas para determinar la cantidad óptima de neuronas por capas, podemos nombrar algunas como:

- El número de neuronas debe estar entre la cantidad de entradas y la cantidad de salidas de la red.

- El número de neuronas debe ser 2/3 del tamaño de la cantidad de entradas más la cantidad de salidas de la red.

- El número de neuronas no debe superar el doble de la cantidad de entradas de la red.

2.2. Marco Teórico Médico

En nuestro estudio de las redes neuronales para predecir accidentes cardíacos es indispensable comparar los resultados obtenidos con otros métodos utilizados actualmente, es por ello que describiremos a continuación uno de los más utilizados, el Coeficiente de Framingham[23], que es un estudio cardiovascular a largo plazo a los residentes de la ciudad de Framingham, Massachusetts. El estudio comenzó en 1948 con 5209 sujetos adultos de Framingham, y se encuentra actualmente en su tercera generación de participantes.
Detallaremos un método de comparación de pruebas que se utiliza con frecuencia en las ciencias médicas, el cálculo de coeficientes, como la Sensibilidad, la Especificidad y el Valor predictivo de una prueba. Este método comparativo nos permite evaluar los resultados obtenidos en las predicciones de las redes neuronales como con el método del Coeficiente de Framingham.

2.2.1. Coeficiente de Framingham

En las últimas décadas se han desarrollado distintas ecuaciones para evaluar el riesgo cardiovascular de un individuo que incluyen los principales factores aterogénicos[1], como las basadas en el estudio americano de Framingham. Nos centraremos en Framingham por ser el estudio poblacional de más años de seguimiento y que más información ha proporcionado sobre los factores de riesgo cardiovascular y su papel predictivo de episodios coronarios.

Entre las distintas ecuaciones para el cálculo del riesgo cardiovascular la desarrollada por los investigadores del Framingham Heart Study es la que ha tenido mayor difusión. Desde la publicación inicial de Kannel W.B., McGee D. y Gordon T.[23] hasta la versión actual, la tabla ha sufrido diversas actualizaciones. La última actualización para hombres es la que aparece en la Figura 2.3 y para mujeres en la Figura 2.4, se pueden encontrar en la tercera revisión del Programa Nacional de Educación sobre el colesterol de Estados Unidos[2].

La ecuación está formada por 6 factores de riesgo: el sexo, la edad, las lipoproteínas de alta densidad (HDL, del inglés High density lipoprotein), el colesterol total, la presión arterial sistólica en reposo y el tabaquismo. A cada factor de riesgo se le asigna una puntuación. La cifra resultante de sumar los puntos obtenidos para cada uno de los 6 factores de riesgo permite establecer el porcentaje de riesgo de sufrir un episodio coronario dentro de los próximos 10 años.

2.2.2. Método Comparativo de Pruebas

Otro método que usaremos para comparar los resultados de las redes, además del error de validación, será la matriz de confusión. En el campo de la inteligencia artificial una matriz de confusión es una herramienta de visualización que se emplea en aprendizaje supervisado. Ésta consiste en una tabla de dos entradas, en el eje horizontal expresamos los resultados de la red neuronal y los comparamos con los resultados esperados colocados en el eje vertical, como observamos en la Figura 2.5.

[1]Def. Conjunto de alteraciones que permiten la aparición en la pared de las arterias de un depósito de lípidos, que finalmente se transformará en una placa de calcificación y facilitará la pérdida de elasticidad arterial y otros trastornos vasculares.

[2]National Cholesterol Education Program, NCEP `http://www.nhlbi.nih.gov/guidelines/cholesterol/risk_tbl.htm`

HOMBRES

Edad	Puntos
20-34	-9
35-39	-4
40-44	0
45-49	3
50-54	6
55-59	8
60-64	10
65-69	11
70-74	12
75-79	13

HDL	Puntos
60+	-1
50-59	0
40-49	1
<40	2

Presión sistólica	Sin tratamiento	Con tratamiento
<120	0	0
120-129	0	1
130-139	1	2
140-159	1	2
160+	2	3

Colesterol	20-39	40-49	50-59	60-69	70-79
<160	0	0	0	0	0
160-199	4	3	2	1	0
200-239	7	5	3	1	0
240-279	9	6	4	2	1
280+	11	8	5	3	1

	20-39	40-49	50-59	60-69	70-79
No Fumador	0	0	0	0	0
Fumador	8	5	3	1	1

Total Puntos	Riesgo a 10 años
< 0	< 1%
0	1%
1	1%
2	1%
3	1%
4	1%
5	2%
6	2%
7	3%
8	4%
9	5%
10	6%
11	8%
12	10%
13	12%
14	16%
15	20%
16	25%
>=17	>=30%

Figura 2.3: Tabla Framingham para hombres

MUJERES

Edad	Puntos
20-34	-7
35-39	-3
40-44	0
45-49	3
50-54	6
55-59	8
60-64	10
65-69	12
70-74	14
75-79	16

HDL	Puntos
60+	-1
50-59	0
40-49	1
<40	2

Presión sistólica	Sin tratamiento	Con tratamiento
<120	0	0
120-129	1	3
130-139	2	4
140-159	3	5
160+	4	6

Colesterol	20-39	40-49	50-59	60-69	70-79
<160	0	0	0	0	0
160-199	4	3	2	1	1
200-239	8	6	4	2	1
240-279	11	8	5	3	2
280+	13	10	7	4	2

	20-39	40-49	50-59	60-69	70-79
No fumador	0	0	0	0	0
Fumador	9	7	4	2	1

Total Puntos	Riesgo a 10 años
< 9	< 1%
9	1%
10	1%
11	1%
12	1%
13	2%
14	2%
15	3%
16	4%
17	5%
18	6%
19	8%
20	11%
21	14%
22	17%
23	22%
24	27%
>=25	>=30%

Figura 2.4: Tabla Framingham para mujeres

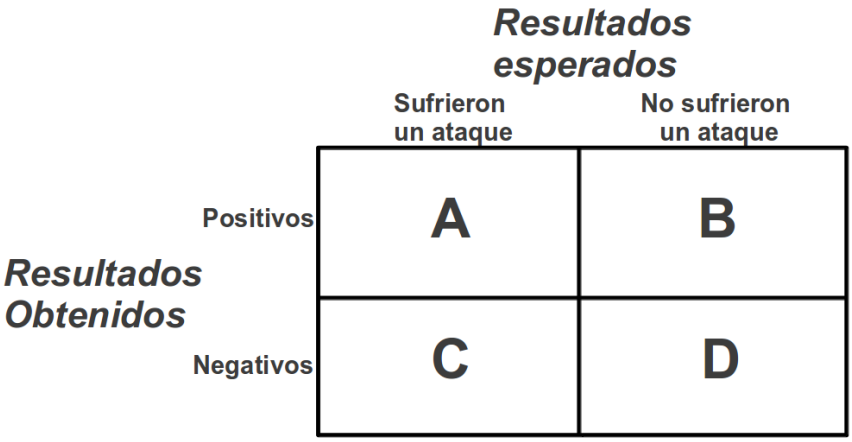

Figura 2.5: Tabla de cuatro casillas

Los valores A+C y B+D respresentan respectivamente el total de pacientes que sufrieron un ataque y el total de pacientes que no lo sufrieron; A+B y C+D representan respectivamente el total de los que van a sufrir el ataque y los que no van a sufrir un ataque según la predicción de la red neuronal. Con estos valores podemos calcular tres indicadores útiles.

1. **Sensibilidad:** es la capacidad de detectar a los que sufrieron realmente un ataque.

$$Sensibilidad = 100\frac{A}{A+C} \tag{2.6}$$

2. **Especificidad:** es la capacidad de detectar a los que no sufrieron un ataque.

$$Especificidad = 100\frac{D}{B+D} \tag{2.7}$$

3. **Valor predictivo:** representa la cantidad de pacientes que aparecen como positivos en la predicción de la red neuronal y que han sufrido realmente un ataque.

$$ValorPredictivo = 100\frac{A}{A+B} \tag{2.8}$$

Esta última fórmula para calcular el **Valor Predictivo** sólo puede ser usada cuando los casos positivos y negativos están en proporción con la prevalencia de la enfermedad. Es por esto que, en nuestro caso, utilizaremos para el calculo del **Valor**

Predictivo el Teorema de Bayes[21].

$$P(E/positivo) = \frac{P(positivo/E)P(E)}{P(positivo/E)P(E) + P(positivo/E')P(E')} \tag{2.9}$$

Como nos muestra la Fórmula 2.9, donde $P(E/positivo)$ es la probabilidad de tener la afección dado el hecho de ser positivo a la prueba, o sea: el valor predictivo de la misma; $P(positivo/E)$ es la probabilidad de ser positivo a la prueba dado el hecho haber tenido la afección, dicho de otra manera, la sensibilidad de la prueba; $P(E)$ es la probabilidad de tener la afección, la prevalencia de la enfermedad o afección; $P(positivo/E')$ es la probabilidad de ser positivo a la prueba, dado el hecho de no tener o no sufrir la afección, o sea 1 menos la especificidad de la prueba; $P(E')$ es la probabilidad de no estar enfermo.

Capítulo 3

Desarrollo

3.1. Origen y Descripción del Banco de Datos

El banco de datos que utilizaremos en nuestra investigación fue cedido por la empresa Lammovil SA. Es una empresa latinoamericana con sede en la ciudad de Córdoba, Argentina. Sus productos están dirigidos a brindar soporte a las diferentes actividades relacionadas a la salud, especialmente de manera remota y descentralizada. Su misión es desarrollar productos tecnológicos innovadores para la atención integral de la salud y brindar servicios de calidad para el sector sanitario. Ésta matriz de datos fue estudiada en la investigación publicada por Spence J. D., Eliasziw M., DiCieco M., Hackam D. G., Galil R. y Lohmann T.[6]. Como mencionamos anteriormente, es un conjunto de datos clínicos y resultados de exámenes médicos realizados a 1686 pacientes canadienses que fueron controlados individualmente por más de 5 años, y grupalmente desde el año 1980 al 2001. Nos centramos en estudiar un conjunto de datos específico, de este seleccionamos un conjunto de variables que detallaremos a continuación:

1. Edad: cantidad de años del paciente.
2. Sexo: femenino o masculino.
3. Peso: expresado en kg.
4. Índice de Masa Corporal: expresado en kg/m^2

5. Diabetes: 0 si no posee, 1 caso contrario.
6. Ataque isquémico transitorio[1]: 0 si no sufrió, 1 caso contrario.
7. Ataque cerebrovascular[2]: 0 si no sufrió 1 caso contrario.
8. Infarto de miocardio[3]: 0 si no sufrió, 1 caso contrario.
9. Paquete de años fumando: cantidad de años que el paciente fumó en su vida.
10. Año en que dejó de fumar.
11. Colesterol: expresado en $mmol/l$[4].
12. Triglicéridos: expresado en $mmol/l$.
13. Lipoproteína de alta densidad: expresado en $mmol/l$.
14. Lipoproteína de baja densidad: expresado en $mmol/l$.
15. Presión Sistólica: expresado en $mmHg$.
16. Presión Diastólica: expresado en $mmHg$.
17. Tratamiento reducción de lípidos: 0 si no está en tratamiento, 1 caso contrario.
18. Tratamiento antihipertensivo: 0 si no está en tratamiento, 1 caso contrario.
19. Placa carotidea[5]: expresado en cm^2.

[1]Es un accidente cerebrovascular de tipo isquémico. Se produce por la falta de aporte sanguíneo a una parte del cerebro, de forma transitoria.

[2]Es la pérdida brusca de funciones cerebrales causada por una alteración vascular, ya sea por interrupción del flujo sanguíneo o por hemorragia.

[3]Riego sanguíneo insuficiente, con daño tisular, en una parte del corazón, producido por una obstrucción en una de las arterias coronarias, frecuentemente por ruptura de una placa de ateroma vulnerable.

[4]En química, la concentracion molar (también llamada molaridad), es una medida de la concentración de un soluto en una disolución, o de alguna especie molecular, iónica, o atómica que se encuentra en un volumen dado expresado en moles por litro, en este caso usamos milimol por litro como unidad de medida

[5]Es la acumulación de una sustancia grasa en las paredes de la arteria carótida forma lo que denominamos placa carotidea, esta se mide según su área en cm^2 y este dato se obtiene a través de una ecografía.

20. Derecha interior: hace referencia al pico máximo de frecuencia en la Ecografía Doppler en la carótida interior derecha. Medida en MHz.

21. Derecha exterior: hace referencia al pico máximo de frecuencia en la Ecografía Doppler en la carótida exterior derecha. Medida en MHz.

22. Derecha común: hace referencia al pico máximo de frecuencia en la Ecografía Doppler en la carótida común derecha. Medida en MHz.

23. Izquierda interior: hace referencia al pico máximo de frecuencia en la Ecografía Doppler en la carótida interior izquierda. Medida en MHz.

24. Izquierda exterior: hace referencia al pico máximo de frecuencia en la Ecografía Doppler en la carótida exterior izquierda. Medida en MHz.

25. Izquierda común: hace referencia al pico máximo de frecuencia en la Ecografía Doppler en la carótida común izquierda. Medida en MHz.

26. Estenosis: 0 si sufre de una estenosis[6] menor al 50 %, 1 caso contrario.

Utilizamos estas variables como conjunto de entrada para las redes neuronales que creamos, con el objetivo de que sean capaces de predecir si el paciente sufrirá alguna de estas tres afeciones:

1. Ataque isquémico transitorio.

2. Ataque cerebrovascular.

3. Infarto de miocardio.

Como contamos con un seguimiento de estos pacientes(más de 200 variables), tenemos el dato si sufrieron o no alguno de estos ataques y así podremos validar la predicción.

[6]En medicina, estenosis o estegnosis es un término utilizado para denotar la constricción o estrechamiento de un orificio o conducto corporal. Puede ser de origen congénito o adquirido por tumores, engrosamiento o hipertrofia , o por infiltración y fibrosis de las paredes o bordes luminales o valvulares.

3.2. Análisis de Datos

La primera actividad realizada fue el análisis del banco de datos, que consiste en un conjunto de resultados de exámenes médicos practicados a pacientes de diferentes edades y sexos de una población determinada. Los pacientes fueron monitoreados durante 5 años calendario, en los cuales se documentó si sufrieron o no un ataque isquémico transitorio, un ataque cerebrovascular o un infarto de miocardio. El banco de datos base constaba de un total de 1689 registros de pacientes, el grupo de pacientes se encuentra entre los 15 años a los 95 años de edad, de nacionalidad canadiense, con una cantidad mayor a 200 variables por cada registro, estas variables representan los distintos datos clínico-médicos de cada paciente, no son datos públicos y fueron utilizados para el desarrollo de otros estudios[6]. Mediante reuniones con especialistas[7], seleccionamos un conjunto de variables influyentes en el resultado que queremos predecir, así se determinaron un total de 26 variables para conformar una base de datos definitiva. Luego de esta decisión, eliminamos los registros que no poseían datos en alguna de las variables elegidas, reduciendo la cantidad de registros a 1076. Para manejar los datos de una mejor manera, se cargaron en una base de datos MySQL[2].

3.3. Normalización de Datos

Como segunda actividad, normalizamos los datos de las variables seleccionadas. La normalización es proporcional a los valores máximos y mínimos, se utilizó la Fórmula 3.1, donde el m y M representan al mínimo y al máximo valor de toda la base de datos, respectivamente, de la variable en cuestión.

$$N(x) = \frac{x - m}{M} \tag{3.1}$$

Para realizar esta tarea utilizamos el lenguaje de programación SQL y las herramientas de MySQL.

[7]Dr. Luis Armando, Dr. Hernan Perez y Dr. Hugo Villafae

3.4. Selección del Banco de Datos Entrenamiento y Validación

Luego de la normalización, siguió la comparación entre diferentes redes. Para esto fue necesario crear diferentes bancos de datos de entrenamiento y de validación. El proceso consistió en seleccionar del banco de datos central un conjunto aleatorio para el entrenamiento y otro para la validación. Dependiendo del escenario el porcentaje de cada banco con respecto al central fue variando y lo veremos en detalle en la sección **Escenarios, entrenamiento y resultados**.

3.5. Investigación de Frameworks

Una vez conformados los bancos de datos, investigamos distintos frameworks para la construcción, entrenamiento y verificación de las redes neuronales. Luego de una investigación superficial a partir de nuestras necesidades, decidimos profundizar el estudio de 2 frameworks. El primero fue el denominado FANN[3](Fast Artificial Neural Network), que es una biblioteca gratis, libre y de código abierto desarrollada en Ç". El segundo fue el denomidado Encog 3[13], que es también una biblioteca desarrollada en Java gratis, libre y de código abierto. En el Cuadro 3.1 observamos algunas características de los dos frameworks estudiados.

Finalmente, decidimos utilizar el framework Encog 3, dejando de lado la velocidad de entrenamiento y de cálculo del primer framework. Esta decisión estuvo influenciada por la conveniencia de este framework para luego ser integrado a una aplicación Android.

3.6. Escenarios, Entrenamiento y Resultados

En esta sección, describiremos los escenarios con las distintas redes creadas y los resultados obtenidos. Los escenarios se conforman por diferentes configuraciones, tales como el conjunto de datos de entrenamiento y validación, tipo de red, algoritmo de entrenamiento, función de activación, configuración de la red y cota de error para el fin del entrenamiento. Como resultado de cada escenario, obtuvimos una red

	FANN	Encog 3
Lenguaje desarrollado	C	Java
Velocidad de procesamiento	Alta	Baja
Configurable	Si	Si
Buena documentación	Si	Si
Multiplataforma	Si	Si
Interfaz gráfica	Si	Si
Libre y de Código abierto	Si	Si
Algoritmos de Aprendizaje	RPROP, Quickprop, Batch, Incremental	SCG, RPROP, BPROP, QPROP, MPROP, Genetic Algorithm, Simulated Annealing, ADALINE, SVD,PSO

Cuadro 3.1: Cuadro Comparativo

entrenada con un de error de validación, que utilizaremos para comparar las distintas redes.

En nuestro estudio definimos a toda la población como los 1076 casos, de éstos sólo 105 son los que efectivamente sufrieron alguna de las tres afecciones antes nombradas, por lo tanto la prevalencia adoptada es del 9.75 %.

Los escenarios elegidos fueron construidos por medio de distintas modificaciones en la configuración de las redes y la cota de error de entrenamiento que utilizamos de corte. En la Figura 3.1 vemos la oganización final de los escenarios elegidos.

Para definir estas configuraciones nos basamos en la teoría descripta en el capítulo anterior, para ello utilizamos por ejemplo en la designación de la estructura (cantidad de neuronas por capas) la regla que nos dice que el número de neuronas en la capa oculta debe ser 2/3 de la cantidad de entradas más la cantidad de salidas de la red y la regla que dice que no debe ser mayor que el doble de la cantidad de entradas.

Escenario 1

Conjunto de datos de entrenamiento seleccionado al azar 600 registros normalizados, conjunto de datos de validación 476 registros normalizados, el tipo de red feedforward, el algoritmo de entrenamiento resilient backpropagation, la función de activación sigmoidal, la topología de la red es (26 entradas, 18 neuronas ocultas, 1

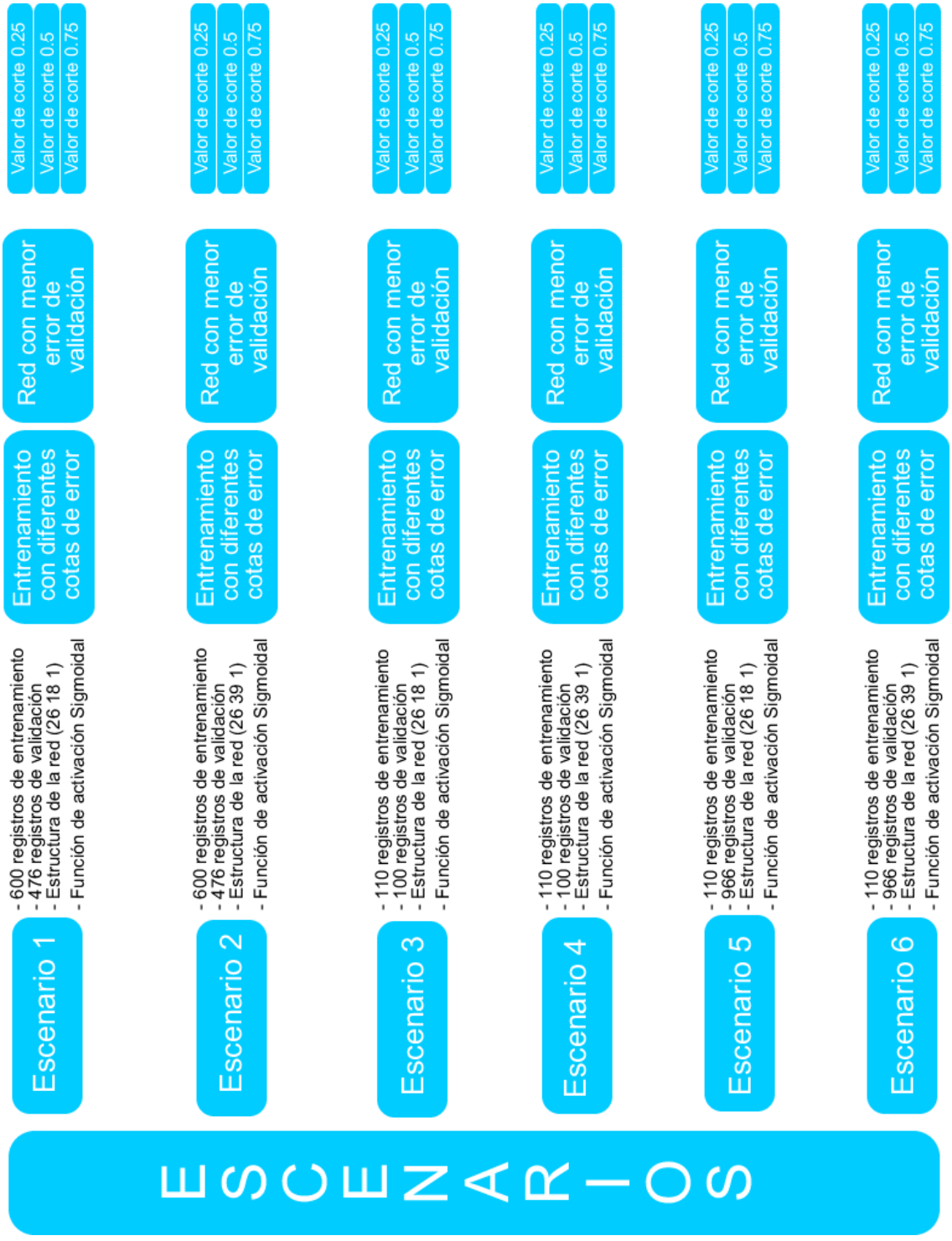

Figura 3.1: Organización de los escenarios

Cota de Error	Error de Entrenamiento	Error de Validación
0.1 %	0.09 %	10.50 %
1 %	0.99 %	10.43 %
2 %	1.98 %	8.61 %
3 %	2.99 %	8.78 %
4 %	3.98 %	8.39 %
5 %	4.99 %	7.85 %
6 %	5.99 %	7.30 %
7 %	6.96 %	7.64 %
8 %	7.93 %	8.05 %
9 %	8.91 %	8.76 %
10 %	9.24 %	16.83 %

Cuadro 3.2: Cuadro de Errores

	Sufrieron un ataque	No sufrieron un ataque
Positivo	21	31
Negativo	25	399

Cuadro 3.3: Tabla de cuatro casillas corte en 0.25

neurona de salida) totalmente conectada. Con ésta configuración de la red, entrenamos a la misma con distintas cotas de error y obtuvimos los siguientes resultados, como muestra el Cuadro 3.2.

Luego del entrenamiento y la validación observamos los resultados y elegimos la red con menor error de validación en este caso la red con cota de error de 6 %, en el Gráfico 3.2 observamos la evolución del entrenamiento logrado de un error de 5.99 % y un error de validación del 7.30 % en un total de 30 iteraciones.

Construimos el Cuadro de cuatro casillas según el valor de corte seleccionado para definir el resultado de la red. Para esto utilizamos 3 tipos de cortes 0.25, 0.5 y 0.75.

En el Cuadro 3.3 observamos el resultado para el valor de corte en 0.25.

Con el Cuadro 3.3 calculamos la Sensibilidad que fue del 45.65 %, la Especificidad del 92.79 % y el Valor Predictivo del 40.62 %.

En el Cuadro 3.4 observamos el resultado para el valor de corte en 0.5.

Con el Cuadro 3.4 calculamos la Sensibilidad que fue del 13.04 %, la Especificidad del 98.37 % y el Valor Predictivo del 46.39 %.

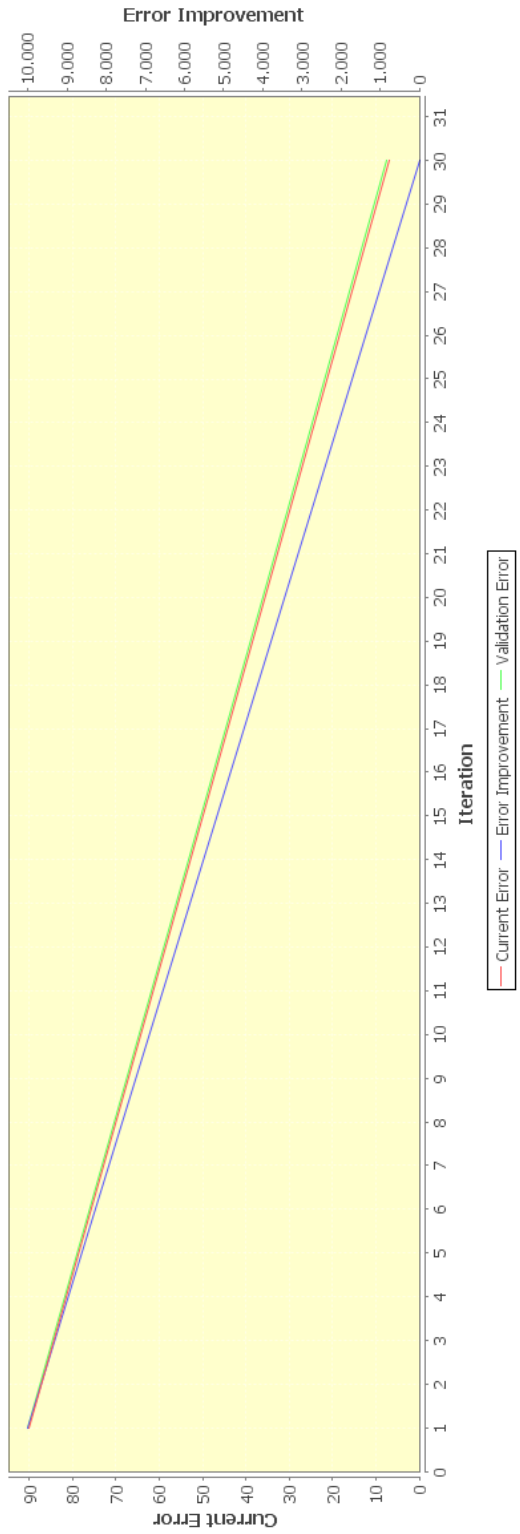

Figura 3.2: Evolución del entrenamiento

	Sufrieron un ataque	No sufrieron un ataque
Positivo	6	7
Negativo	40	423

Cuadro 3.4: Tabla de cuatro casillas corte en 0.5

	Sufrieron un ataque	No sufrieron un ataque
Positivo	5	0
Negativo	41	430

Cuadro 3.5: Tabla de cuatro casillas corte en 0.75

En el Cuadro 3.5 observamos el resultado para el valor de corte en 0.75.

Con el Cuadro 3.5 calculamos la Sensibilidad que fue del 10.86 %, la Especificidad del 100 % y el Valor Predictivo del 100 %.

Escenario 2

Idem escenario 1 solo modificamos la estructura de la red que fue de (26 entradas, 39 neuronas ocultas, 1 neurona de salida).

Con ésta configuración de la red, entrenamos a la misma con distintas cotas de error y obtuvimos los siguientes resultados, como muestra el Cuadro 3.6.

Luego del entrenamiento y la validación observamos los resultados y elegimos la red con menor error de validación en este caso la red con cota de error de 5 %, en el Gráfico 3.3 observamos la evolución del entrenamiento logrado de un error de 4.98 % y un error de validación del 7.07 % en un total de 106 iteraciones.

En el Cuadro 3.7 observamos el resultado para el valor de corte en 0.25.

Con el Cuadro 3.7 calculamos la Sensibilidad que fue del 54.34 %, la Especificidad del 91.62 % y el Valor Predictivo del 41.22 %.

En el Cuadro 3.8 observamos el resultado para el valor de corte en 0.5.

Con el Cuadro 3.8 calculamos la Sensibilidad que fue del 26.08 %, la Especificidad del 98.37 % y el Valor Predictivo del 63.68 %.

En el Cuadro 3.9 observamos el resultado para el valor de corte en 0.75.

Con el Cuadro 3.9 calculamos la Sensibilidad que fue del 10.86 %, la Especificidad del 99.76 % y el Valor Predictivo 83.46 %.

Cota de Error	Error de Entrenamiento	Error de Validación
0.1 %	0.09 %	8.42 %
1 %	0.99 %	8.27 %
2 %	1.99 %	7.70 %
3 %	2.99 %	7.39 %
4 %	3.99 %	7.61 %
5 %	4.98 %	7.07 %
6 %	5.97 %	7.28 %
7 %	6.94 %	7.74 %
8 %	7.92 %	8.10 %
9 %	8.64 %	9.02 %
10 %	9.70 %	71.51 %

Cuadro 3.6: Cuadro de Errores

	Sufrieron un ataque	No sufrieron un ataque
Positivo	25	36
Negativo	21	394

Cuadro 3.7: Tabla de cuatro casillas corte en 0.25

	Sufrieron un ataque	No sufrieron un ataque
Positivo	12	7
Negativo	34	423

Cuadro 3.8: Tabla de cuatro casillas corte en 0.5

	Sufrieron un ataque	No sufrieron un ataque
Positivo	5	1
Negativo	41	429

Cuadro 3.9: Tabla de cuatro casillas corte en 0.75

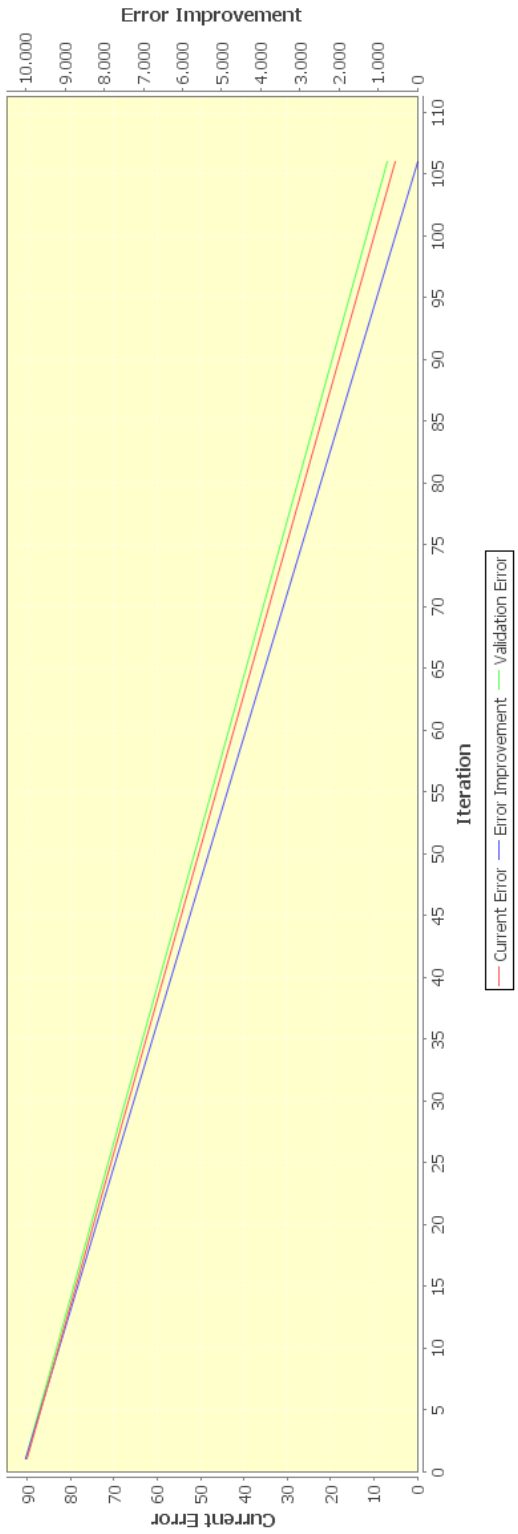

Figura 3.3: Evolución del entrenamiento

Cota de Error	Error de Entrenamiento	Error de Validación
0.1 %	0.09 %	22.09 %
1 %	0.98 %	18.23 %
2 %	1.99 %	17.00 %
3 %	2.98 %	17.93 %
4 %	3.96 %	18.04 %
5 %	4.97 %	17.56 %
10 %	9.85 %	14.06 %
15 %	14.67 %	13.49 %
17 %	16.97 %	13.97 %
20 %	19.41 %	16.55 %
30 %	25.62 %	44.84 %

Cuadro 3.10: Cuadro de Errores

Escenario 3

En este escenario modificamos la cantidad de datos de entrenamiento y validación así como también el porcentaje de los casos positivos y negativos, el conjunto de datos de entrenamiento es de 110 registros normalizados donde hay 50 % de casos positivos y 50 % de casos negativos, el conjunto de datos de validación es de 100 registros normalizados con la misma proporción de casos negativos y positivos. Los registros se seleccionaron al azar. El tipo de red es feedforward, el algoritmo de entrenamiento es resilient backpropagation, la función de activación es sigmoidal, la topología de la red es de (26 neuronas de entrada, 18 neuronas ocultas, 1 neurona de salida) totalmente conectada.

Con ésta configuración de la red, entrenamos a la misma con distintas cotas de error y obtuvimos los siguientes resultados, como muestra el Cuadro 3.10.

Luego del entrenamiento y la validación observamos los resultados y elegimos la red con menor error de validación en este caso la red con cota de error de 15 %, en el Gráfico 3.4 observamos la evolución del entrenamiento logrado de un error de 14.67 % y un error de validación del 13.49 % en un total de 31 iteraciones.

En el Cuadro 3.11 observamos el resultado para el valor de corte en 0.25.

Con el Cuadro 3.11 calculamos la Sensibilidad que fue del 98 %, la Especificidad del 57.99 % y el Valor Predictivo del 20.13 %.

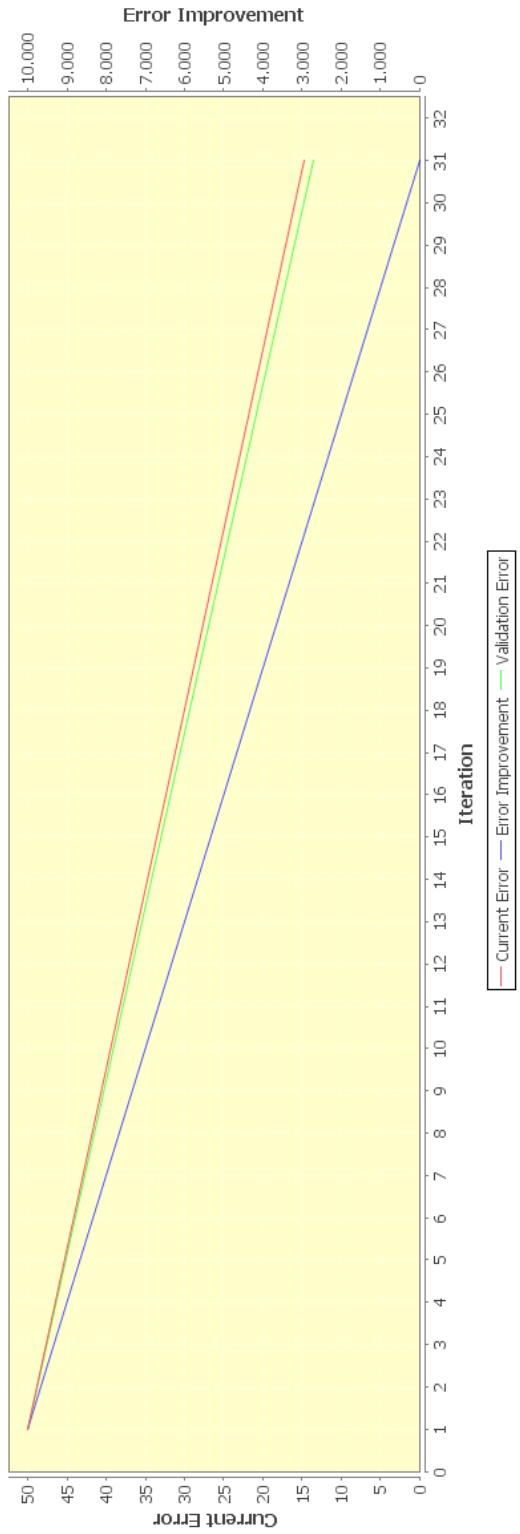

Figura 3.4: Evolución del entrenamiento

	Sufrieron un ataque	No sufrieron un ataque
Positivo	49	21
Negativo	1	29

Cuadro 3.11: Tabla de cuatro casillas corte en 0.25

	Sufrieron un ataque	No sufrieron un ataque
Positivo	42	11
Negativo	8	39

Cuadro 3.12: Tabla de cuatro casillas corte en 0.5

En el Cuadro 3.12 observamos el resultado para el valor de corte en 0.5.

Con el Cuadro 3.12 calculamos la Sensibilidad que fue del 84 %, la Especificidad del 78 % y el Valor Predictivo del 29.20 %.

En el Cuadro 3.13 observamos el resultado para el valor de corte en 0.75.

Con el Cuadro 3.13 calculamos la Sensibilidad que fue del 50 %, la Especificidad del 94 % y el Valor Predictivo del 47.37 %.

Escenario 4

Idem escenario 1 solo modificamos la estructura de la red que fue de (26 entradas, 39 neuronas ocultas, 1 neurona de salida).

Con ésta configuración de la red, entrenamos a la misma con distintas cotas de error y obtuvimos los siguientes resultados, como muestra el Cuadro 3.14.

Luego del entrenamiento y la validación observamos los resultados y elegimos la red con menor error de validación en este caso la red con cota de error de 15 %, en el Gráfico 3.5 observamos la evolución del entrenamiento logrado de un error de 14.81 % y un error de validación del 14.53 % en un total de 35 iteraciones.

En el Cuadro 3.15 observamos el resultado para el valor de corte en 0.25.

Con el Cuadro 3.15 calculamos la Sensibilidad que fue del 96 %, la Especificidad

	Sufrieron un ataque	No sufrieron un ataque
Positivo	25	3
Negativo	25	47

Cuadro 3.13: Tabla de cuatro casillas corte en 0.75

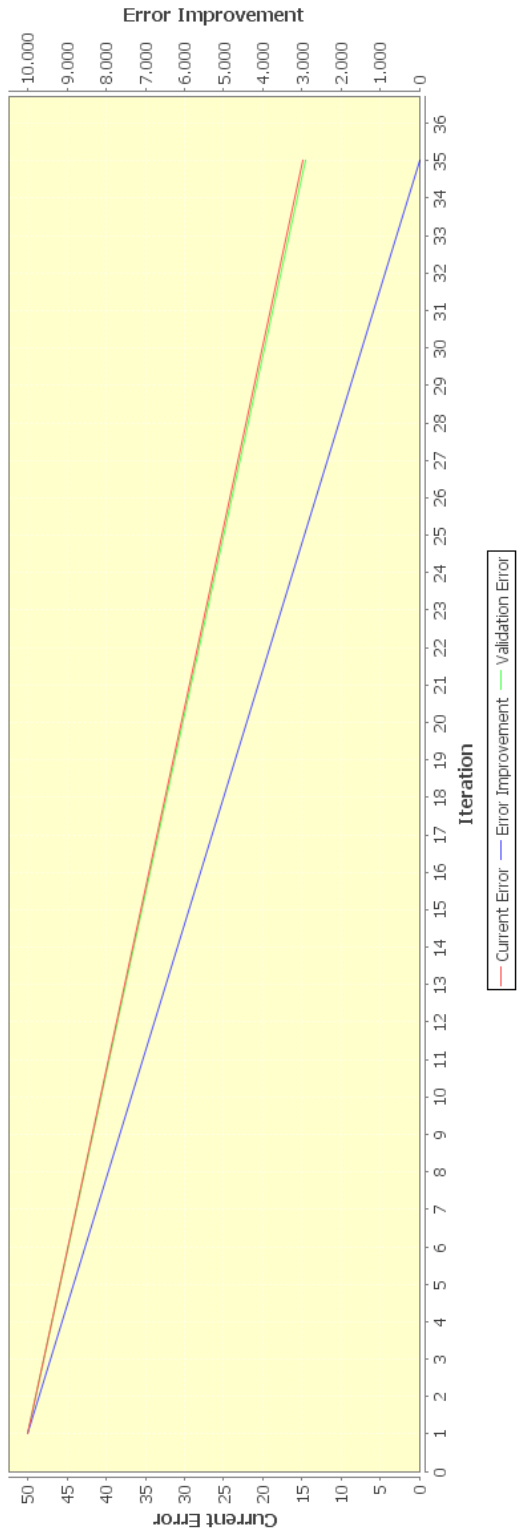

Figura 3.5: Evolución del entrenamiento

Cota de Error	Error de Entrenamiento	Error de Validación
0.1 %	0.09 %	19.80 %
1 %	0.99 %	16.78 %
2 %	1.94 %	15.74 %
3 %	2.98 %	15.15 %
4 %	3.97 %	15.15 %
5 %	4.99 %	14.90 %
10 %	9.97 %	14.75 %
13 %	12.95 %	14.56 %
15 %	14.81 %	14.53 %
17 %	16.77 %	15.18 %
20 %	19.70 %	17.07 %

Cuadro 3.14: Cuadro de Errores

	Sufrieron un ataque	No sufrieron un ataque
Positivo	48	25
Negativo	2	25

Cuadro 3.15: Tabla de cuatro casillas corte en 0.25

del 50 % y el Valor Predictivo del 17.17 %.

En el Cuadro 3.16 observamos el resultado para el valor de corte en 0.5.

Con el Cuadro 3.16 calculamos la Sensibilidad que fue del 84 %, la Especificidad del 74 % y el Valor Predictivo del 25.87 %. En el Cuadro 3.17 observamos el resultado para el valor de corte en 0.75.

Con el Cuadro 3.17 calculamos la Sensibilidad que fue del 52 %, la Especificidad del 94 % y el Valor Predictivo del 48.35 %.

Escenario 5

Idem escenario 3 modificando el conjunto de datos de validación a 966 registros.

Con ésta configuración de la red, entrenamos a la misma con distintas cotas de

	Sufrieron un ataque	No sufrieron un ataque
Positivo	42	13
Negativo	8	37

Cuadro 3.16: Tabla de cuatro casillas corte en 0.5

	Sufrieron un ataque	No sufrieron un ataque
Positivo	26	3
Negativo	24	47

Cuadro 3.17: Tabla de cuatro casillas corte en 0.75

Cota de Error	Error de Entrenamiento	Error de Validación
0.1 %	0.09 %	30.00 %
1 %	0.96 %	27.74 %
5 %	4.95 %	26.37 %
7 %	6.81 %	23.92 %
10 %	9.97 %	22.02 %
11 %	10.98 %	22.41 %
13 %	12.85 %	25.57 %
15 %	14.76 %	26.31 %
17 %	16.78 %	19.78 %
18 %	16.78 %	20.15 %

Cuadro 3.18: Cuadro de Errores

error y obtuvimos los siguientes resultados, como muestra el Cuadro 3.18.

Luego del entrenamiento y la validación observamos los resultados y elegimos la red con menor error de validación en este caso la red con cota de error de 17 %, en el Gráfico 3.6 observamos la evolución del entrenamiento logrado de un error de 16.78 % y un error de validación del 19.78 % en un total de 61 iteraciones.

En el Cuadro 3.19 observamos el resultado para el valor de corte en 0.25.

Con el Cuadro 3.19 calculamos la Sensibilidad que fue del 98 %, la Especificidad del 39.51 % y el Valor Predictivo del 14.89 %.

En el Cuadro 3.20 observamos el resultado para el valor de corte en 0.5.

Con el Cuadro 3.20 calculamos la Sensibilidad que fue del 82 %, la Especificidad del 70.74 % y el Valor Predictivo del 23.24 %. En el Cuadro 3.21 observamos el resultado para el valor de corte en 0.75.

	Sufrieron un ataque	No sufrieron un ataque
Positivo	49	554
Negativo	1	362

Cuadro 3.19: Tabla de cuatro casillas corte en 0.25

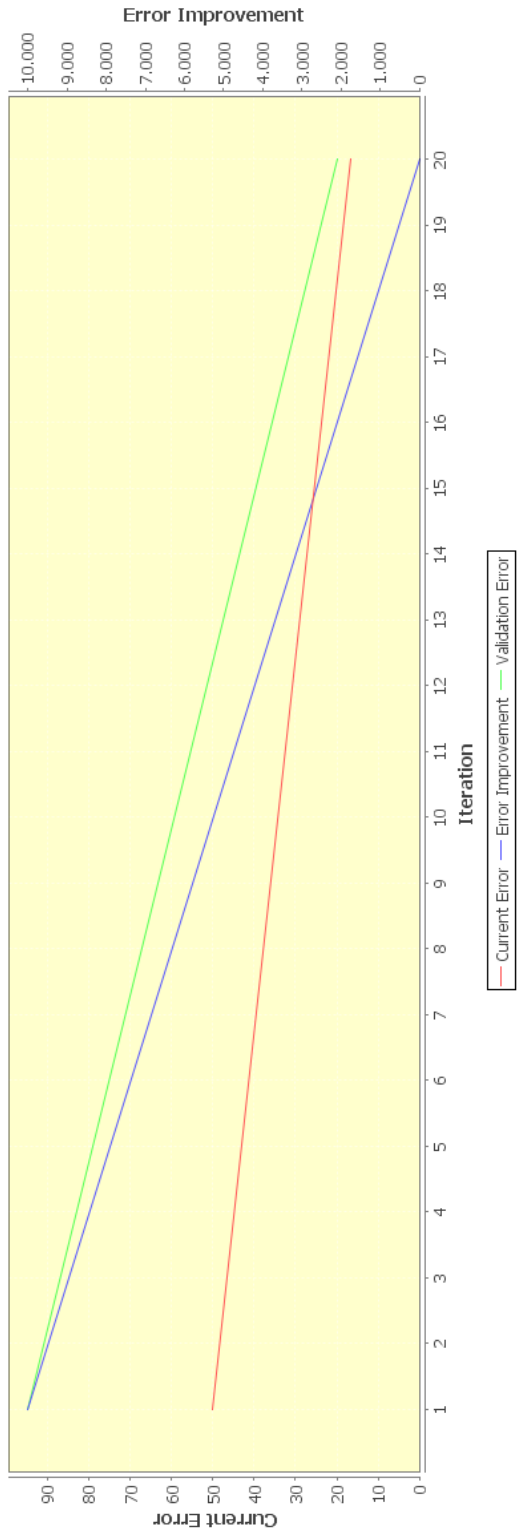

Figura 3.6: Evolución del entrenamiento

	Sufrieron un ataque	No sufrieron un ataque
Positivo	41	268
Negativo	9	648

Cuadro 3.20: Tabla de cuatro casillas corte en 0.5

	Sufrieron un ataque	No sufrieron un ataque
Positivo	26	104
Negativo	24	812

Cuadro 3.21: Tabla de cuatro casillas corte en 0.75

Con el Cuadro 3.21 calculamos la Sensibilidad que fue del 52 %, la Especificidad del 88.64 % y el Valor Predictivo del 33.10 %.

Escenario 6

Idem escenario 5 solo modificamos la estructura de la red que fue de (26 entradas, 39 neuronas ocultas, 1 neurona de salida).

Con ésta configuración de la red, entrenamos a la misma con distintas cotas de error y obtuvimos los siguientes resultados, como muestra el Cuadro 3.22.

Luego del entrenamiento y la validación observamos los resultados y elegimos la red con menor error de validación en este caso la red con cota de error de 17 %, en el Gráfico 3.7 observamos la evolución del entrenamiento logrado de un error de 16.97 % y un error de validación del 18.47 % en un total de 26 iteraciones.

En el Cuadro 3.23 observamos el resultado para el valor de corte en 0.25.

Cota de Error	Error de Entrenamiento	Error de Validación
0.1 %	0.09 %	26.27 %
1 %	0.97 %	25.18 %
5 %	4.96 %	22.73 %
10 %	9.97 %	20.20 %
13 %	12.95 %	19.93 %
15 %	14.72 %	19.44 %
17 %	16.97 %	18.47 %
20 %	19.73 %	24.69 %

Cuadro 3.22: Cuadro de Errores

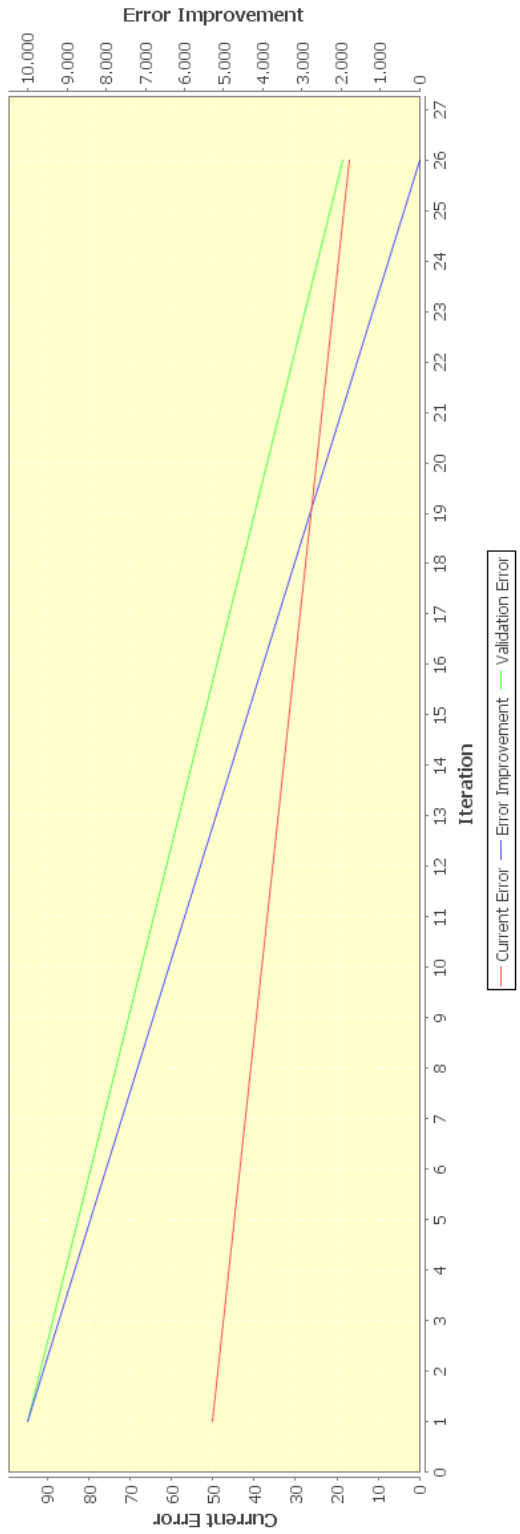

Figura 3.7: Evolución del entrenamiento

	Sufrieron un ataque	No sufrieron un ataque
Positivo	49	552
Negativo	1	364

Cuadro 3.23: Tabla de cuatro casillas corte en 0.25

	Sufrieron un ataque	No sufrieron un ataque
Positivo	40	258
Negativo	10	658

Cuadro 3.24: Tabla de cuatro casillas corte en 0.5

Con el Cuadro 3.23 calculamos la Sensibilidad que fue del 98 %, la Especificidad del 39.73 % y el Valor Predictivo del 14.94 %.

En el Cuadro 3.24 observamos el resultado para el valor de corte en 0.5.

Con el Cuadro 3.24 calculamos la Sensibilidad que fue del 80 %, la Especificidad del 71.83 % y el Valor Predictivo del 23.48 %. En el Cuadro 3.25 observamos el resultado para el valor de corte en 0.75.

Con el Cuadro 3.25 calculamos la Sensibilidad que fue del 46 %, la Especificidad del 91.26 % y el Valor Predictivo del 36.26 %.

Para poder comparar estos resultados calculamos el Coeficiente de Framingham para los bancos de datos de validación de las redes. Utilizamos los datos de validación de los escenarios 1 y 2, el de los escenarios 3 y 4, y por ultimo el de los escenarios 5 y 6. Considerando a los valores mayores a 15 en el Coeficiente de Framingham como predicción positiva, o un alto nivel de riesgo cardíaco, obtubimos como resultado lo que muestra el Cuadro 3.26.

Con el Cuadro 3.26 calculamos la Sensibilidad que fue del 71.56 %, la Especificidad del 56.04 % y el Valor Predictivo del 14.98 %.

Lo mismo realizamos con el banco de datos de los escenarios 3 y 4. En el Cuadro 3.27 observamos los resultados para este banco de datos.

Con el Cuadro 3.27 calculamos la Sensibilidad que fue del 68 %, la Especificidad

	Sufrieron un ataque	No sufrieron un ataque
Positivo	23	80
Negativo	27	836

Cuadro 3.25: Tabla de cuatro casillas corte en 0.75

	Sufrieron un ataque	No sufrieron un ataque
Positivo	33	189
Negativo	13	241

Cuadro 3.26: Resultados Predicción con Framingham

	Sufrieron un ataque	No sufrieron un ataque
Positivo	34	19
Negativo	16	31

Cuadro 3.27: Resultados Predicción con Framingham

del 62 % y el Valor Predictivo del 16.20 %.

Lo mismo realizamos con el banco de datos de los escenarios 5 y 6. En el Cuadro 3.28 observamos los resultados para este banco de datos.

Con el Cuadro 3.28 calculamos la Sensibilidad que fue del 68 %, la Especificidad del 57 % y el Valor Predictivo del 14.61 %.

En el Cuadro 3.29 observamos los resultados obtenidos en todos los escenarios de pruebas realizados y los de Framingham.

	Sufrieron un ataque	No sufrieron un ataque
Positivo	34	393
Negativo	16	523

Cuadro 3.28: Resultados Predicción con Framingham

Escenarios	Valor de Corte	Error Validación	Sensibilidad	Especificidad	Valor Predictivo
1	0.25	7.30 %	41.30 %	90.46 %	31.87 %
1	0.5	7.30 %	13.04 %	98.37 %	46.39 %
1	0.75	7.30 %	10.86 %	100 %	100 %
2	0.25	7.07 %	54.34 %	91.62 %	41.22 %
2	0.5	7.07 %	26.08 %	98.37 %	41.22 %
2	0.75	7.07 %	10.86 %	99.76 %	83.46 %
3	0.25	13.49 %	98 %	57.99 %	15.14 %
3	0.5	13.49 %	84 %	78 %	29.20 %
3	0.75	13.49 %	50 %	94 %	47.37 %
4	0.25	14.53 %	96 %	50 %	17.17 %
4	0.5	14.53 %	84 %	74 %	25.87 %
4	0.75	14.53 %	52 %	94 %	48.35 %
5	0.25	19.78 %	98 %	39.51 %	14.89 %
5	0.5	19.78 %	82 %	70.74 %	23.24 %
5	0.75	19.78 %	52 %	88.64 %	33.10 %
6	0.25	18.47 %	98 %	39.73 %	14.94 %
6	0.5	18.47 %	80 %	71.83 %	23.48 %
6	0.75	18.47 %	46 %	91.26 %	36.26 %
Framingham 1	-	-	71.56 %	56.04 %	14.98 %
Framingham 2	-	-	68 %	62 %	16.20 %
Framingham 3	-	-	68 %	57 %	14.61 %

Cuadro 3.29: Tabla de resultados de los distintos Escenarios

3.7. Desarrollo de Aplicación Android

Con las redes neuronales entrenadas y verificadas, desarrollamos una aplicación en la plataforma Android[1]. Este programa consiste en una interfaz de usuario simple, un formulario donde se cargan las variables que la red neuronal necesita para ejecutarse. Cuenta con un botón que ejecuta la red neuronal y nos devuelve el resultado de la misma. El diseño de la interfaz de usuario se ve en la figura 3.8. Para poder realizar esta aplicación nos basamos en la biblioteca Encog 3 desarrollada en Java. Realizamos un tratamiento en los datos previos a la ejecución de la red y devolvemos el resultado obtenido. Como la red neuronal ya esta entrenada el costo computacional de esta aplicación es lineal.

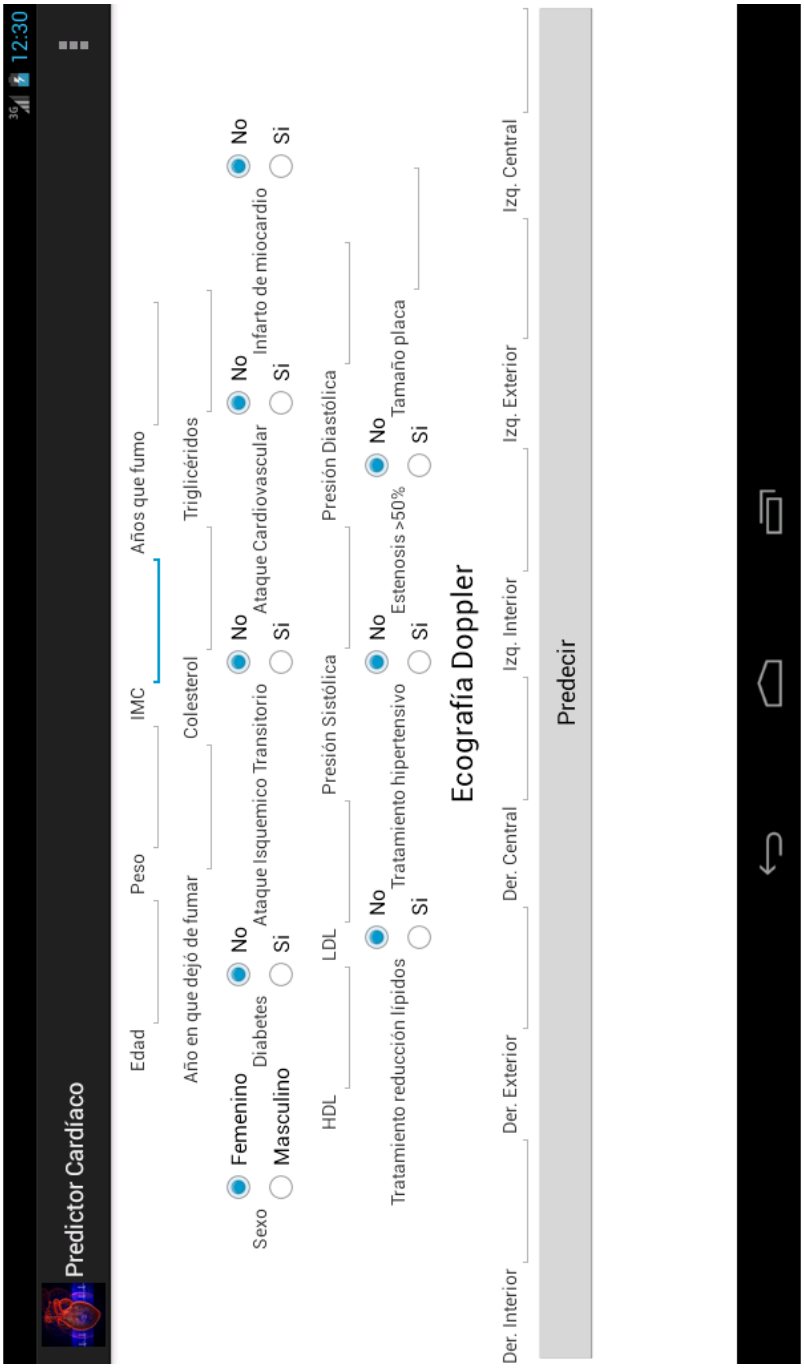

Figura 3.8: Interfaz de Aplicación Android

Capítulo 4

Conclusiones

4.1. Análisis de Resultados

Uno de nuestros objetivos era construir una red neuronal artificial capaz de predecir accidentes cardíacos de manera confiable y con mejores resultados que el método conocido como coeficiente de Framingham. Además de contribuir a partir de una rama las Ciencias de la Computación a problemas de la Medicina, favoreciendo no sólo un diálogo interdisciplinario, sino también la resolución de interrogantes actuales de otras ciencias. Observando los resultados obtenidos, cumplimos con estos objetivos. Encontramos tres escenarios en donde superamos los resultados obtenidos con la metodología de Framingham. Estos escenarios fueron:

1. El escenario 3 con valor de corte en 0.5 obtuvo un valor de 84 % de sensibilidad, un 78 % de especificidad y un 29.20 % de valor predictivo.

2. El escenario 4 con valor de corte en 0.5 obtuvo un valor de 84 % de sensibilidad, un 74 % de especificidad y un 25.87 % de valor predictivo.

3. El escenario 5 con valor de corte en 0.5 obtuvo un valor de 82 % de sensibilidad, un 70.74 % de especificidad y un 23.24 % de valor predictivo.

4. El escenario 6 con valor de corte en 0.5 obtuvo un valor de 80 % de sensibilidad, un 71.83 % de especificidad y un 23.48 % de valor predictivo.

Mientras que con la metodología de Framingham obtuvimos:

1. Con el conjunto de datos de validación del escenario 1 y 2, obtuvo un valor de 71.56 % de sensibilidad, un 56.04 % de especificidad y un 14.98 % de valor predictivo.

2. Con el conjunto de datos de validación del escenario 3 y 4, obtuvo un valor de 68 % de sensibilidad, un 62 % de especificidad y un 16.20 % de valor predictivo.

3. Con el conjunto de datos de validación del escenario 5 y 6, obtuvo un valor de 68 % de sensibilidad, un 57 % de especificidad y un 14.61 % de valor predictivo.

Vale destacar que para este método sólo utilizamos un subconjunto de 8 variables de las 26 que requiere la red neuronal desarrollada. Aunque esto podría verse como una desventaja, se debe tener en cuenta que el coeficiente de Framingham es un estudio estadístico de una población significante, que exige contar con una gran cantidad de datos. En cambio, con el resultado obtenido en este estudio se demuestra que con un conjunto de 100 casos para entrenamiento de una red se obtienen mejores resultados que con el método de Framingham. Esto hace más factible la aplicación en poblaciones donde no se cuenta con tantos datos para realizar el estudio estadístico de Framingham.

4.2. Trabajos Futuros

Como trabajos futuros prevemos diferentes actividades. Una de estas es la investigación centrada en averiguar qué variables, de las utilizadas, tienen una mayor incidencia en la predicción. Para este trabajo podemos realizar distintas pruebas y estudios estadísticos de los resultados obtenidos en la investigación. Por otro lado seguir investigando el rendimiento de nuevos modelos de redes neuronales y compararlos con los resultados obtenidos de los modelos estudiados. Por último y no menos importante utilizar un banco de datos con historias clínicas de pacientes argentinos para reproducir lo generado con el banco de datos canadiense y así construir una herramienta que se adapte con mayor presición a las características de la población argentina.

Bibliografía

[1] *Android Developers*. URL http://developer.android.com/develop/index.html.

[2] *MySQL 5.0 Reference Manual*. URL http://dev.mysql.com/doc/refman/5.0/es/.

[3] *Reference Manual for FANN 2.2.0*. URL http://leenissen.dk/fann/html/files/fann-h.html.

[4] Widrow B. y Hoff M. E. Adaptive switching circuits. *IRE WESCON Convention Record*, (4):96–104, 1960.

[5] Michie D., Spiegelhalter D.J., y Taylor C.C. *Machine Learning*. Neural and Statistical Classification, 1994.

[6] Spence J. D., Eliasziw M., DiCieco M., Hackam D. G., Galil R., y Lohmann T. *Carotid plaque area a tool for targeting and evaluating vascular preventive therapy*. Stroke, 2002.

[7] Rasmussen C. E. y Williams C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[8] Rumelhart D. E., Hinton G. E., y Williams R. J. Learning representations by back-propagating errors. *Nature (London)*, (323):533–536, 1986.

[9] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review Vol*, (65):386–408, 1958.

[10] Rosenblatt F. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Spartan Books, 1962.

[11] Cybenko G. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, (2):303–314, 1989.

[12] Heaton J. *Introduction to neural networks with Java*. Heaton Research, Inc., 2008.

[13] Heaton J. *Programming neural networks with encog 3 in Java*. Heaton Research, Inc., 2011.

[14] Werbos P. J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Tesis Doctoral, Harvard University, 1974.

[15] Minsky M. L. y Papert S. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

[16] Sackett D. L., Haynes R. B., Guayatt G. H., y Tugwell P. *Epidemiología clínica ciencia básica para la medicina clínica*. Editorial Medica Panamericana, 1994.

[17] Bishop C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1996.

[18] Mitchell T. M. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997.

[19] Duda R. O., Hart P. E., y Stork D. G. *Pattern Clasification*. Wiley-Interscience, 2000.

[20] McCulloch W. S. y Pitts W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, (5):115–133, 1943.

[21] Bayes T. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, (53):370–418, 1963.

[22] Guerrero R. V., Gonzales C. L., y Medina E. L. *Epidemiología*. Addison-Wesley Iberoamericana, 1986.

[23] Kannel W.B., McGee D., y Gordon T. A general cardiovascular risk profile: the framingham study. *Am J Cardiol*, (38):46–51, 1976.