UNIVERSIDAD DE PANAMÁ

VICERRECTORÍA DE INVESTIGACIÓN Y POSTGRADO

PROGRAMA DE MAESTRÍA EN MATEMÁTICA

MODELIZACIÓN DE LA PREDICCIÓN DE RIESGO DE LA DINÁMICA DE TRANSMISIÓN DE LA INFLUENZA EN LA PROVINCIA DE PANAMÁ

JULIO E. TRUJILLO G.

TESIS PRESENTADA COMO UNO DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE MAESTRÍA EN MATEMÁTICA CON OPCIÓN EN INVESTIGACIÓN DE OPERACIONES

PANAMÁ, REPÚBLICA DE PANAMÁ

2020

Agradecimiento

«La Matemática posee no sólo la verdad, sino la suprema belleza, una belleza fría y austera, como la de una escultura.»

Bertrand Russell.

Primero quiero agradecer al mejor matemático, el creador de este universo a «Dios», por toda la fortaleza que me ha dado para culminar esta maestría y por darme sabiduría que se encuentra encerrada en el «Libro».

Segundo, agradezco el apoyo de «Mamita» y «Papito», por apoyarme económicamente durante mí formación académica y por la educación que me dieron.

Tercero, también doy gracias a todos los profesores que estuvieron involucrados en mí formación, especialmente a mí directora de tesis a la profesora Iveth Martínez.

Por último, agradezco a la Universidad de Panamá, por reunir a todos los profesores que algún momento me dieron clases.

Índice general

Índice de figuras

Índice de cuadros

Resumen

Una de las aplicaciones de los autómatas celulares es modelizar la propagación de una enfermedad en una población y de las redes neuronales es de clasificación en clases. Encontraremos un modelo para la dinámica de la influenza en una población y un modelo de clasificación de pacientes con cuadros graves de COVID-19 usando redes neuronales.

Summary

One of the applications of cellular automata is to model the spread of a disease in a population and of neural networks is to classify them. We will find a model for the dynamics of influenza in a population and a model for classifying patients with severe COVID-19 conditions using neural networks.

Introducción

El poco conocimiento de los principios de la modelización matemática y el tratamiento de datos, son algunas de las limitantes al momento de modelar un fenómeno natural o físico. Su valor para analizar situaciones reales e interpretar resultados ha crecido notablemente de modo que, al aplicar los modelos en el caso específico de epidemias, resulta ser una herramienta de apoyo favorable por las siguientes razones:

1. Revelan patrones que no son obvios.
2. Es posible obtener información de las relaciones existente entre los distintos elementos que intervienen.
3. Es posible utilizar los modelos para predecir el comportamiento de las enfermedades, como fundamento para la toma decisiones y así ejecutar acciones para el cuidado de una población.

Dada la importancia que involucra el conocimiento del comportamiento de ciertas enfermedades, el propósito del presente trabajo es desarrollar un modelo matemático de la dinámica de transmisión de la influenza mediante las reglas que reflejen su comportamiento con autómatas celulares y redes neuronales.

Existen, en la actualidad, múltiples modelos de pronóstico desarrollado para enfermedades que utilizan autómatas celulares, como: *Modelamiento Computacional de la Dinámica de Transmisión de la Varicela mediante Autómatas Celulares (Cell-DEVS)*

(Romero y col., 2018), *A Model Based on Cellular Automata to Si-mulate Epidemic Diseases* (White y col., 2006) y *Dynamic Cellular Automata Based Epidemic Spread Model for Population in Patches with Movement* (Athitan y col., 2014).

Para lograr un modelo sea que capaz de pronosticar la dinámica de la influenza fue necesario:

1. Estudiar el riesgo de la influenza y las políticas públicas para su prevención y control con información proporcionada por el Sistema de Vigilancia Epidemiológica de la República de Panamá.
2. Describir un patrón del comportamiento de infección de la influenza en la población.
3. Relacionar las reglas de comportamiento de contagio con autómatas celulares.
4. Simular la transmisión de la influenza en una población, con el uso de los autómatas celulares.
5. Establecer las relaciones entre las distintas variables que conlleve al análisis del riesgo de contagio en Panamá
6. Determinar distintas estrategias que puedan reducir la tasa de transmisión de la influenza.

El estudio y los resultados obtenidos se plasman en este documento organizado en tres (3) capítulos. En El primer capítulo se expone los fundamentos básicos de los autómatas celulares y las redes neuronales, importantes para el entendimiento de la investigación, el segundo capítulo contiene el modelo basado en autómatas celulares que modela el comportamiento de una enfermedad en una población y además el modelo de las redes neuronales que clasifica las personas que podrían complicarse dados algunos síntomas previos. En tercero se implementa el prototipo desarrollado, la experimentación del modelo y los principales resultados con-

seguidos y finalmente, las conclusiones del trabajo, las contribuciones realizadas y las recomendaciones para todas aquellas personas interesadas en el tema o que deseen continuar con investigaciones en esta área.

Capítulo 1

Preliminares

Un autómata celular es un modelo matemático que se describe como una colección de objetos simples que interactúan uno con otro, proporcionando una información adecuada de un sistema natural. Para mejor compresión de este tema, abordaremos conceptos y propiedades de esta herramienta matemática.

1.1. Autómatas Celulares

En lo que sigue se definirá autómatas celulares de dimensión uno y la definición para dimensión superior se derivada de ella.

Sea $S = \{0, 1, \cdots k-1\}$ el conjunto de estado de cada elemento o célula, que llamaremos alfabeto. La secuencia bilateral

$$X = \cdots x_{-2}\ x_{-1}\ x_0\ x_1\ x_2 \cdots$$

es llamada la configuración del autómata celular y al conjunto de todas las configuraciones es conocido como el espacio de configuración, $\Sigma = S^{\mathbb{Z}}$, donde $\mathbb{Z}$ es el conjunto de los números enteros.

Para nuestro caso, se considerará que el tiempo es discreto y que cada configuración, de todas células cambian a la vez.

Para cada tiempo, el cambio de estado de una célula depende de su estado actual, y de los $2r$ vecinos alrededor de él y por la función

$$f : S_t^{2r+1} \to S_{t+1}$$

de la forma

$$x_i^{t+1} = f(x_{i-r}^t, \cdots, x_i^t, \cdots, x_{i+r}^t) \tag{1.1}$$

La dinámica de la evolución del autómata celular está determinada por una función F de cada célula individual, es decir,

$$F : \Sigma \to \Sigma$$

sobre la configuración total del autómata celular que de manera general está dada por

$$\Omega^{(t+1)} = F\Omega^{(t)} \subseteq \Omega^{(t)}$$

donde $\Omega^{(t)} = F^t\Sigma$.

Definición 1.1 *La función $F : \Sigma \to \Sigma$ es llamada autómata celular (AC) para la cual existen $r > 0$ y $f : S_t^{2r+1} \to S_{t+1}$ que permiten que cualquiera configuración de la forma*

$X = \cdots x_{-2}\ x_{-1}\ x_0\ x_1\ x_2 \cdots$ *que satisfaga*

$$F(X)_i = f(x_{i-r}, \cdots, x_i, \cdots, x_{i+r})$$

1.2. Partes del autómata celular

Los autómatas celulares están compuestos de células, mallas, vecinos, regla evolutiva y los estados de las células.

1.2.1. Células

Son la unidad más básica de un autómata celular y son distribuidas sobre una malla en un espacio Euclidiano.

1.2.2. Malla

También conocida como espacio celular, es un conjunto de puntos en un espacio que representa la distribución de las células. El espacio celular puede dividirse en una malla $d - dimensional$ de cualquier espacio.

Ejemplo **1.1** *En el plano de dimensión dos, un autómata celular consiste en una malla de $m \times n$ cuadrados donde cada cuadrado, son las células.*

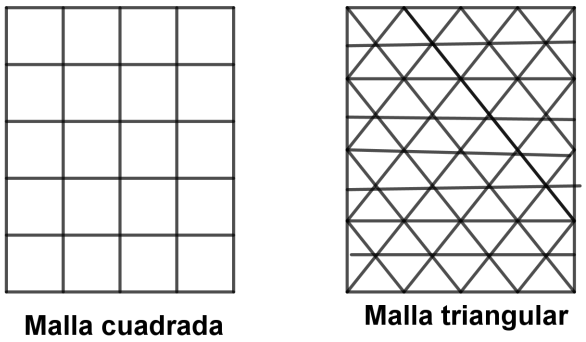


Figura 1.1: Mallas

En lo que sigue consideramos nuestro espacio celular como una malla cuadrada, ya que es intuitiva y más simple, lo que resulta ser más adecuada para el desarrollo computacional. También, tiene desventajas, ya que no modela el fenómeno de isotropía.

1.2.3. Condiciones de frontera

Técnicamente, se considera el espacio celular con una extensión infinita en todas las dimensiones. Pero en la práctica se suele implementar límite y condiciones de

frontera, como:

i) Frontera periódica o circular. Es aquel espacio celular en las cuales sus fronteras opuestas están conectadas, en dimensión dos se puede visualizar como un toro.

ii) Frontera reflectora. Las células que están en el exterior toman el estado de las células del interior, como si fuera un espejo.

iii) Frontera constante o abierta. Todas las células del exterior tienen un estado.

1.2.4. Configuración inicial

Es la distribución y combinación de todos los estados de las células en el espacio celular inicial del sistema.

1.2.5. Vecindad

Es un conjunto parcial de células alrededor de una célula e incluyéndose, que necesita para determinar el siguiente estado.

i) Dimensión uno.

Es determinada por un radio r. Todas las células que están a un radio r de una célula específica son consideradas como vecinas.

ii) Dimensión dos.

Su definición es más complicada, pero usualmente se tomará alguna de las siguientes tipos:

i) Vecindad de von Neumann

Son las cuatro células que están ubicadas en la vertical y la horizontal

adyacente a la célula. En este caso el radio es igual a uno.

$$\begin{aligned}\mathcal{N}_N^1(i,j) &= \{x_{ij} : |i-k| + |j-l| \leq 1\} \\ &= \{x_{ij}, x_{i-1j}, x_{ij-1}, x_{i+1j}, x_{ij+1}\}\end{aligned}$$

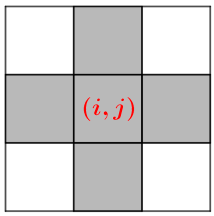


Figura 1.2: Vecindad de von Neumann

ii) Vecindad de Moore

Se puede considerar está vecindad como una extensión de la von Neumann, ya que se incluye las cuatro células que están en las diagonales.

$$\begin{aligned}\mathcal{N}_M^1(i,j) &= \{x_{kl} : |i-k| \leq 1, |j-l| \leq 1\} \\ &= \mathcal{N}_N^1(i,j) \cup \{x_{i-1j-1}, x_{i-1j+1}, x_{i+1j-1}, x_{i+1j+1}\}\end{aligned}$$

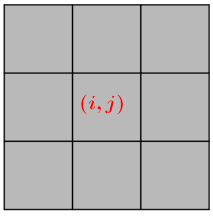


Figura 1.3: Vecindad de Moore

iii) Vecindad extendida de Moore

Para este tipo de vecindad se toma el radio igual a 2 o más grande.

1.2.6. Regla evolutiva

Una función dinámica determina el estado de la célula en el siguiente momento, que depende del estado actual de la célula y de sus vecinos. Es decir,

$$S_{x_{ij}}^{t+1} = f(S_{x_{ij}}^{t}, S_{\mathcal{N}(i,j)}^{t})$$

donde $S_{x_{ij}}^{t}$ representa el estado de la célula x_{ij} en el momento t y $S_{\mathcal{N}(i,j)}^{t}$ es el conjunto de estados de las células en la vecindad $\mathcal{N}(i,j)$ de la célula x_{ij} y f es una función que representa un conjunto de reglas de transición.

Ejemplo **1.2** *Sea* $r = 1$ *el radio de vecindad y* $k = 2$ *el número de estados, donde* $S = \{0, 1\}$. *La función* f *se define por*

$$x_i^{t+1} = f(x_{i-1}^t, x_i^t, x_{i+1}^t)$$

Como $x_{i-1}^t, x_i^t, x_{i+1}^t$ *pueden tener dos estados posibles, existen* 2^3 *combinaciones posibles, descritos de la forma.*

Input	111	110	101	100	011	010	001	000
Output	0	1	0	0	1	1	0	0

Cuadro 1.1: Regla 76

El valor x_i^1 en la posición i en la configuración $A^{(1)} = FA^{(0)} = \Omega^{(1)}$ depende de una vecindad de tres lugares $\{x_{i-1}^0, x_i^0, x_{i+1}^0\}$ en la configuración anterior $A^{(0)} = \Omega^{(0)}$. El x_{i+1}^1 depende de la vecindad superpuesta $\{x_i^0, x_{i+1}^0, x_{i+2}^2\}$. La dependencia de x_{i+1}^1 sobre x_i^0 asociada con estos dos lugares se puede representar mediante un grafo G (Figura 1.4).

Los nodos en el grafo G representan las superposiciones $\{x_i^0, x_{i+1}^0\}$. Estos nodos están unidos por arcos dirigidos correspondientes a vecindades de tres lugares. La regla en el cuadro 1.1 define una transformación para cada vecindad de tres lugares y, por lo tanto, asocia un símbolo con cada arco de G.

Cada posible camino a través de G corresponde a una configuración inicial particular $A^{(0)}$. El sucesor $A^{(1)}$ de cada configuración inicial viene dada por la secuencia de símbolos asociados con los arcos en el camino. La secuencia de símbolos obtenidas siguiendo todos los caminos posibles a través de G corresponden así a todas las configuraciones posibles $A^{(1)}$ obtenidas después de un paso de tiempo en la evolución del autómata celular.

Así, el conjunto completo $\Omega^{(1)}$ puede representarse mediante el grafo G.

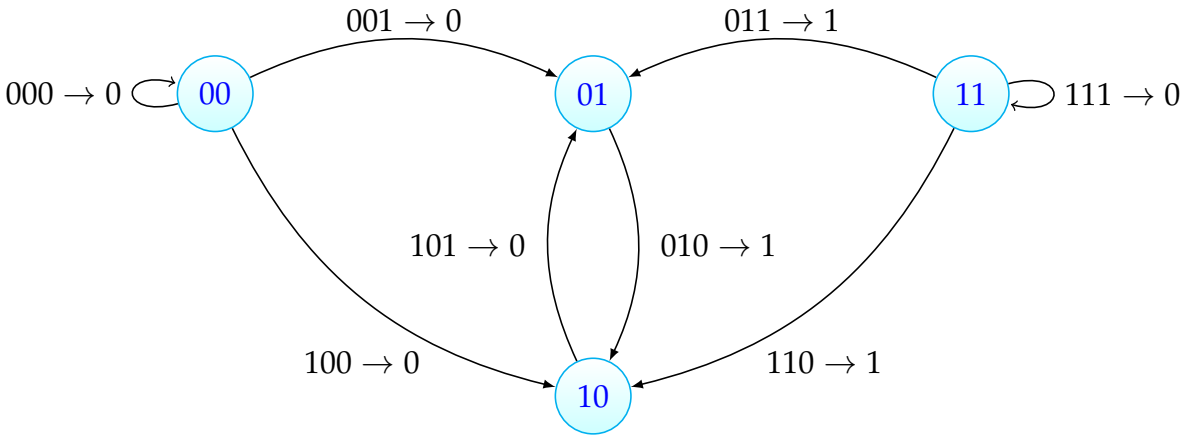


Figura 1.4: Grafo de transición de los estados de la regla 76

Ejemplo **1.3** *Conway's Game of Life es un autómata celular diseñado por el matemático británico John Horton Conway en 1970. Se define por la siguiente función*

$$f(S^t_{x_{ij}}, S^t_{\mathcal{N}(i,j)}) = \begin{cases} 1 & , S^t_{x_{ij}} = 1 \; y \; (\#(S^t_{\mathcal{N}(i,j)}) = 2 \; o \; \#(S^t_{\mathcal{N}(i,j)}) = 3) \\ 1 & , S^t_{x_{ij}} = 0 \; y \; \#(S^t_{\mathcal{N}(i,j)}) = 3 \\ 0 & , \textit{en otro caso.} \end{cases}$$

donde el conjunto $S = \{0, 1\}$ *se describe como: Muerta* $\rightarrow 0$ *y Viva* $\rightarrow 1$.

Sea $\#(\cdot)$ *la función que devuelve la cantidad células con estado igual a 1.*

1.3. Clasificación de los autómatas celulares

En el libro *Cellular Automata and Complexity*, (Wolfram, 1994) propone una clasificación de las reglas de los autómatas celulares en cuatro tipos, teniendo en cuenta que es para los autómatas celulares de una dimensión y, además, considerando la evolución del sistema desde un estado inicial «desordenado»

i) La evolución conduce a un estado homogéneo.

Es lo mismo que la evolución hacia un atractor en un punto fijo en un sistema dinámico.

Las reglas para esta clase de autómatas celulares toman una función f en (1.1) tal que tenga el mismo valor para todos sus k^{2r+1} posibles conjuntos de argumentos.

ii) La evolución conduce a un conjunto de estructura estable o periódicas simples separadas.

Es lo mismo que la evolución hacia el ciclo límite en un sistema dinámico.

iii) La evolución conduce a un patrón caótico.

Esto es lo mismo que evolucionando hacia un atractor extraño en un sistema dinámico.

iv) La evolución conduce a estructura localizadas, a veces de larga duración.

El comportamiento del sistema es completamente aleatorio, ni completamente ordenado.

En el mismo libro se extiende la clasificación para autómatas celulares de dimensión dos, titulado *Two dimensional cellular automata*.

1.4. Topología y dinámica de los Autómata Celulares

Una métrica sobre el espacio de configuraciones $\Sigma = S^{\mathbb{Z}^2}$ es definida por

$$d(X,Y) = \begin{cases} 0 & X = Y \\ 2^{-\min\{||(i,j)||_\infty :\, x_{ij} \neq y_{ij}\}} & X \neq Y \end{cases}$$

donde

$$||(i,j)||_\infty = \max\{|i|,|j|\}$$

La topología que genera $d(\cdot,\cdot)$ es la topología de Cantor.

Sea $X \in \Sigma$ una configuración arbitraria y $D \subseteq \mathbb{Z}^2$ un conjunto finito de células. El cilindro

$$Cilindro(X,D) = \{Y \in \Sigma :\ x_{ij} = y_{ij}, \forall (i,j) \in D\}$$

determinado por X y D que contiene todas las configuraciones que concuerdan con X en el interior de D. Este cilindro es un conjunto abierto y cerrado.

Se establece lo siguiente:

Proposición 1.1 *El conjunto Σ es un espacio métrico compacto y el cilindro forma un conjunto clopen básico. Cada AC la función $F : \Sigma \to \Sigma$ es continua.*

El par (Σ, F) donde Σ es compacto y $F : \Sigma \to \Sigma$ es continua es comúnmente llamado sistema dinámico.

El teorema Curtis-Hedlund-Lyndon de 1969 muestra dos propiedades que caracteriza las funciones F de AC:

Teorema 1.1 *La función $F : \Sigma \to \Sigma$ es un autómata celular si y sólo si es continua y conmuta con traslaciones.*

Corolario 1.1 *Una función F autómata celular es reversible si y sólo si es una biyección.*

1.5. Computabilidad de los autómatas celulares

Los AC proporciona un modelo ideal para estudiar las conexiones entre la dinámica y la computación. Uno de los problemas principales en estudiar los sistemas dinámicos es la «Complejidad», que es un término no bien definido y ambiguo para cuantificar.

En el caso de los ACs, existen distintas medidas como la entropía de la información, información mutua, entre otras, y las mismas son usadas, pero su utilidad es limitada.

Los problemas computacionales se han dividido en varias clases, consideremos tres: **P**, **NP** y **NP** − **complete**.

i) **P**: son problemas resolubles en un tiempo polinomial del tamaño del problema.

Ejemplo **1.4** *Estos problemas son de tipo* **P**

- *Determinar el camino más corto.*
- *Encontrar el sucesor de una secuencia de tamaño n en un AC, demora un tiempo lineal n.*
- *Máximo común divisor.*

ii) **NP**: son problemas resolubles en un tiempo que aumenta más rápido que uno polinomial, pero si tenemos una solución, su verificación se hace en un tiempo polinomial.

Ejemplo **1.5** *Factorizar un entero es un caso de problema* **NP**.

iii) **Np** − **complete**: son problemas **NP**, a los cuales todos los demás problemas **NP** pueden reducirse en tiempo polinomial.

Ejemplo **1.6** *Casos de problemas* **Np** − **complete** *podemos mencionar:*

- *«Satisfiability» (SAT), encontrar valores de verdad para n variables. las cuales hacen verdadera una expresión particular Booleana.*
- *Determinar la preimagen para una secuencia de un cierto autómata celular, o determinar si una secuencia particular es generada alguna vez.*

Observación. El problema de determina si una secuencia en particular es generada en la evolución de un AC es indecidible, lo que significa que no es posible encontrar un algoritmo que lo resuelva en forma general.

1.6. Redes Neuronales

El desarrollo de las ciencias de la computación ha motivado que los modelos matemáticos sean automatizados y utilizados como herramientas para el desarrollo de la humanidad.

Como he de esperarse existen modelos que intentan simular el modo que el ser humano aprende, conocidos como los modelos de Inteligencia artificial (I.A.).

Es así que tomar como modelo una neurona cerebral y abstraer sus partes que permita describir su comportamiento, dio origen a una nueva área de estudio llamada Redes neuronales artificiales o simplemente redes neuronales (RN).

Nuestro punto de partida va ser el modelo propuesto por los fisiólogos y biofísicos ingleses, Alan Lloyd Hodgkin y Andrew Fielding Huxley, que en 1952 describieron los mecanismos iónicos por los que producían y propagaban los pulsos a lo largo del axón gigante de un calamar. Además, explicaremos el modelo presentado por (Berzal, 2019).

El mecanismo del potencial de acción, depende de la apertura de canales de iones, en la membrana celular de la neurona. Esos canales permiten que los iones de sodio, Na^+, penetren el axón y causen un aumento del potencial de la membrana, que al ser de grasa actúa como un aislante, descrito en la fig. 1.5.

En este modelo el potencial de la membrada V_M está dado por

$$C_M \frac{dV_M}{dt} = -g_{Na}(V_M - E_{Na}) - g_K(V_M - E_K) - g_{leak}(V_M - E_{leak})$$

donde

i) g indica la conductividad de los canales de sodio, potasio y pérdida o escape (*leak*).

ii) Los potenciales E corresponde a los gradientes electroquímicos asociados al flujo de los iones: E_{Na} (sodio), E_K potasio y E_{leak} (Corriente perdida).

iii) C_M es la capacidad asociada a la membrana como condensador.

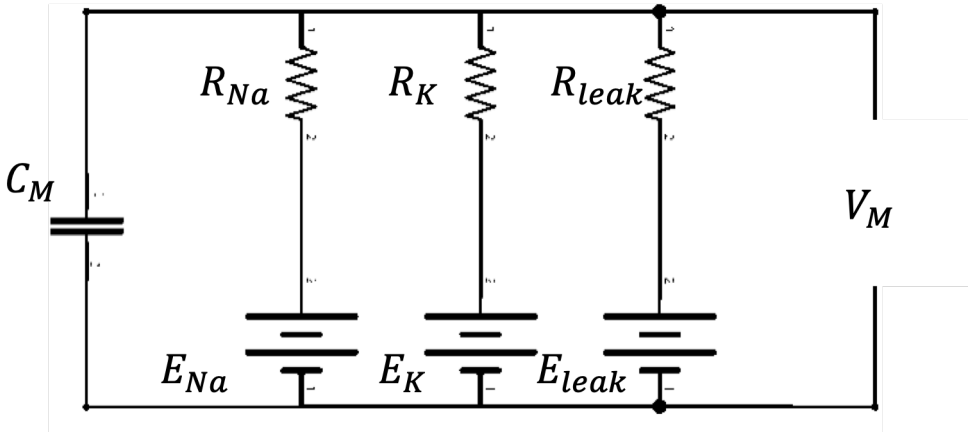


Figura 1.5: Modelo de Hodgkin-Huxley

Cuando se conectan múltiples segmentos del axón, aparecen el fenómeno de la propagación del potencial de acción o pulso (*spike*). En el caso de un circuito el pulso se movería en las dos direcciones, pero en una neurona biológica sólo se movería en un sentido.

Ignorando los canales de iones, obtenemos el siguiente modelo «*Leaky integration and fire neuron model*»

$$C_M \frac{dV_M}{dt} = -g_{leak}(V_j - E_{leak}) + g_{int} \sum_{i=1}^{n} x_i w_{ij}$$

donde

i) V_j representa el potencial de la membrana para la neurona $j - ésima$.

ii) x_i la salida actual de la neurona $j - ésima$.

iii) w_{ij} el peso de la conexión desde la neurona i hasta la neurona j.

Asumiremos que $E_{leak} = 0$, $g_{leak} = 1$, $g_{int} = 1$ y exigiremos que la neurona no tenga memoria, obteniendo

$$V_j = \sum_{i=1}^{n} w_{ij} x_i$$

En ocasiones nos interesa que el comportamiento de la neurona tenga un sesgo, que es similar a las corrientes de pérdidas en una neurona biológica. El sesgo va depender del estado de la neurona, descrito de la forma:

$$\Delta V_j = -V_j + \sum_{i=1}^{n} w_{ij} x_i$$

donde ΔV_j dota a la neurona de una capacidad de memoria. Gran parte de los modelos consideran un sesgo externo o *bias*, b_j

$$V_j = \sum_{i=1}^{n} w_{ij} x_i + b_j$$

De manera que $b_j = w_{0j}$, está asociado a una entrada fija $x_0 = 1$.

En la entrada, una neurona combina las diferentes entradas x_i con sus pesos asociados, para así determinar la entrada total z_j, descrita por

$$z_j = \sum_{i} w_{ij} x_i$$

En la activación, utiliza los valores de entrada para generar una salida y_j, obteniendo un modelo numérico

$$y_j = f(z_j) = f\left(\sum_i w_{ij} x_i\right)$$

donde f es una función de activación, usualmente no lineal. Lo podemos escribir como

$$y_j = f(\mathbf{w}^\mathbf{T}\mathbf{x})$$

donde $\mathbf{w}$ es un vector de pesos w_{ij} y $\mathbf{x}$ el vector de entradas x_j. cuando se toman varias neuronas que interactúan conjuntamente, se forma una capa de neuronas.

La notación vectorial simplificada al diseñar la red neuronal compuesta por varias capas, nos da la expresión.

$$\mathbf{y} = F(W\mathbf{x}) \tag{1.2}$$

donde $\mathbf{y}$ es el vector de salida de la capa ($m \times 1$); $\mathbf{x}$ es el vector de entradas ($n \times 1$) y W es la matriz de pesos ($m \times n$).

Es relativamente sencillo incorporar los sesgos de las neuronas a la expresión (1.2), obteniendo la expresión.

$$\mathbf{y} = F(W\mathbf{x}) + \mathbf{b} \tag{1.3}$$

1.7. Funciones de activación

Como hemos visto las distintas entradas x_i se combinan con sus pesos w_i para obtener la entrada neta z_j, a esta se le puede aplicar una función de activación o transferencia. De esta manera dada una entrada neta que recoge el estímulo, la función de activación es la que determinará la salida de la neurona.

1.7.1. Clasificación de las funciones de activación

Las funciones de activación se pueden clasificar en distintos tipos, dependiendo del tipo de entrada.

i) Funciones de activación discretas. Son aquellas que sólo pueden tomar un conjunto finito de valores.

Es común considerar para este tipo de funciones las de neuronas binarias , $f(z_j) \in \{0,\ 1\}$, o las neuronas bipolares, $f(z_j) \in \{1,\ -1\}$.

Ejemplo **1.7** *Consideremos la función (neuronas bipolares) threshold (o Heaviside o sgn):*

$$f(z) = \begin{cases} 1 & ,z > 0 \\ -1 & ,\textit{en otro caso}. \end{cases}$$

Esto se puede ver como la separación de dos clases en este caso como una recta, dada por la ecuación:

$$w_{1j}x_1 + w_{2j}x_2 + b = 0$$

La capa individual representa una función discriminante y una representación geométrica de la recta umbral, dada por la ecuación

$$x_2 = -\frac{w_{1j}}{w_{2j}}x_1 - \frac{b}{w_{2j}}$$

en que se observa que los pesos determinan la pendiente de la recta y el error dado por $\frac{b}{w_{2j}}$ indicando el «*offset*», es decir, que tan alejado está del origen. En este caso es importante destacar que el vector w es perpendicular a la función discriminante, como se puede ver en la figura 1.6

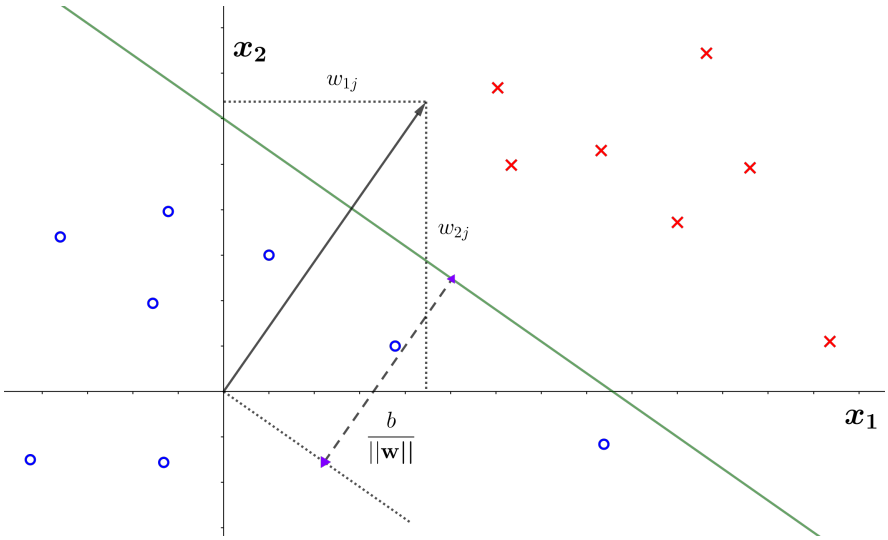


Figura 1.6: Recta umbral

ii) Funciones de activación continuas. Son aquellas que pueden tomar cualquier valor dentro de un intervalo. Es común usar $[0,\ 1]$ o $[-1,\ 1]$ y es fácil cambiarlo, usando

$$f_{[a,\ b]}(z) = a + (b - a) f_{[0,\ 1]}(z)$$

Se clasificas en:

a) Función lineal.

$$f_{lin}(z) = z$$

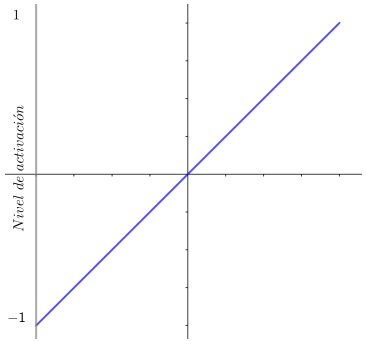


Figura 1.7: Función lineal

b) Función escalón.

La conocida como función de unidad umbral lógico (TLU por *Threshold*

logic units)

$$f_{tlu}(z) = \begin{cases} 1 & ,z \geq 0 \\ 0 & ,z < 0. \end{cases}$$

c) La función signo (*Signum function* o *symetric hard limiter*)

$$f_{sgn}(z) = \begin{cases} 1 & ,z \geq 0 \\ -1 & ,z < 0. \end{cases}$$

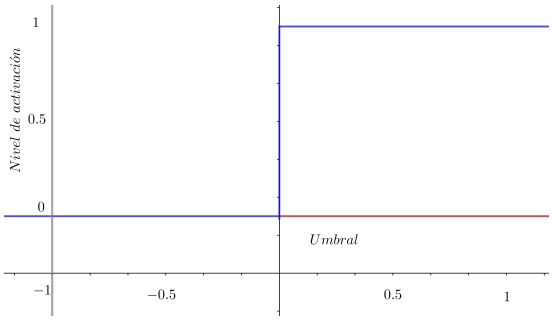


Figura 1.8: Función escalón

d) Función lineal con saturación.

$$f_{sl}(z) = \begin{cases} 1 & ,z > \frac{1}{2} \\ z + \frac{1}{2} & ,-\frac{1}{2} \leq z \leq \frac{1}{2} \\ 0 & ,z < \frac{1}{2} \end{cases}$$

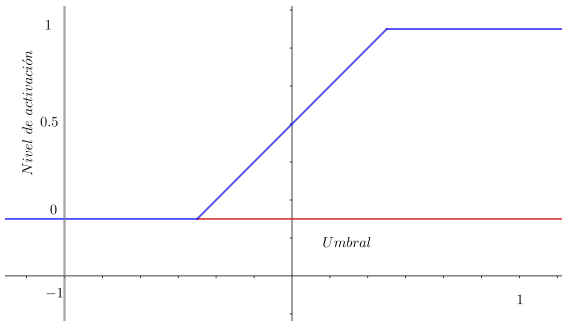


Figura 1.9: Función lineal con saturación

e) Función de activación sigmoidal. Es de gran utilidad cuando deseamos que una función de activación sea no lineal, estrictamente creciente, continua y derivable y tienen forma de «S». Entre este tipo de funciones tenemos:

Función logística

$$f_{logistic}(z) = \frac{1}{1 + e^{-z}}$$

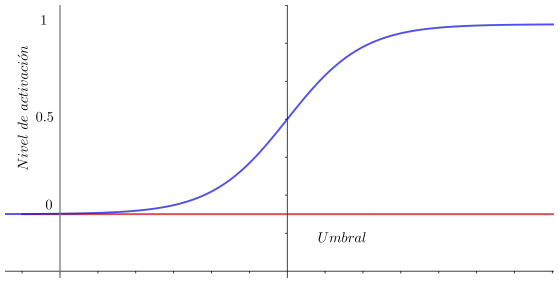


Figura 1.10: Función logística

Función hiperbólica

$$f_{tanh}(z) = \tanh(z)$$

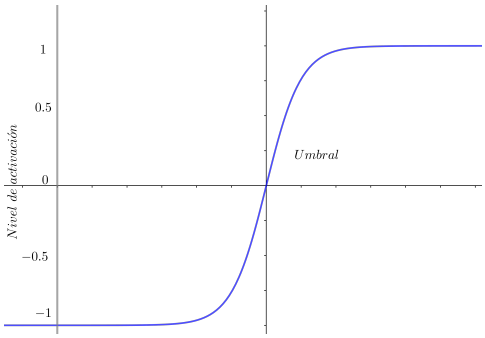


Figura 1.11: Función hiperbólica

Función gudermanniana

$$f_{gd}(z) = \int_0^z \frac{1}{\cosh(t)} dt$$

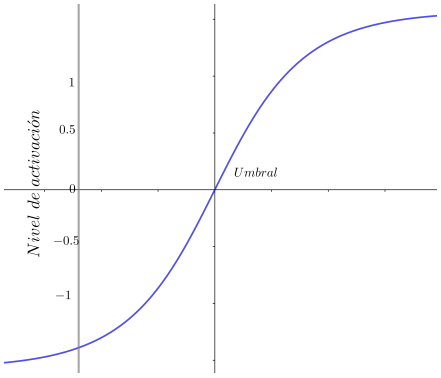


Figura 1.12: Función gudermanniana

f) Función de activación lineal rectificada o ReLu (*Rectified Linear Units*). Son aquellas en la que podemos obtener una función no lineal, sin utilizar funciones trascendentales y se define

$$f_{relu}(z) = \begin{cases} z & , z \geq 0 \\ 0 & , z < 0. \end{cases}$$

1.8. Tipos de aprendizaje en las redes neuronales artificiales

Por la flexibilidad de las RN no es necesario programarlas cuando se cambia el entorno o el ambiente, ajustando su comportamiento adaptándose fácilmente al nuevo ambiente. Esto es posible por la variación de los pesos, que a la vez conlleva al aprendizaje, (Hernández y col., 2004).

Es común clasificar el aprendizaje en: *Aprendizaje Supervisado* o *Aprendizaje no Supervisado*. La diferencia entre ellos, es la existencia de un «supervisor» que controla el aprendizaje.

i) Aprendizaje Supervisado. El aprendizaje con profesor que conoce la respuesta que debería generar RN a partir de una determina entrada.

   La salida de la RN es comparada con la respuesta correcta, y si no coincide se ajusta los pesos hasta que la salida se aproxime a la deseada. Este tipo de aprendizaje es asociado a los problemas de clasificación y regresión.

   Existen tres clases de aprendizaje supervisado:

   1) Corrección de error,

   2) refuerzo y

   3) estocástico.

ii) Aprendizaje no Supervisado. No se cuenta con un supervisor, sólo se proporciona a la RN una entrada y la misma deberá autoorganizarse, en función de algún tipo de estructura existente que presente los datos de entrada. Serán útiles en problemas de clúster y reducción de dimensionalidad.

   Se divide en dos tipos de aprendizaje sin supervisión:

   1) Regla de Hebb y

2) aprendizaje competitivo.

1.9. Perceptrón

El origen del *Deep learning* se remonta a 1943, cuando Warren McCulloch y Walter Pitts describieron en un artículo el primer modelo de red neuronal artificial.

Fueron los primeros en estudiar el cerebro desde un enfoque computacional, en la que tuvieron la habilidad de resolver problemas sin interpretar el funcionamiento del ordenador de forma algorítmica. Fue así, que plantearon un algoritmo de aprendizaje que se ajustara los parámetros de la red automáticamente, como si fuera una máquina de Turing.

Alan Turing, fue el primero en considerar el entrenamiento de redes, conectadas de forma aleatoria como lo demostró en su publicación de 1948 «*Unorganized machine*».

Posteriormente, fueron muchos los investigadores que contribuyeron al desarrollo de la Inteligencia Artificial (I.A.), de los cuales destacamos a Frank Rosenblatt, quien diseñó el algoritmo de aprendizaje del perceptrón (Algorithm 1) y quien utilizó el término conexionista para distinguir su aproximación a I.A.

1.9.1. Regla de aprendizaje del perceptrón

Supongamos que tenemos un conjunto de entrenamiento que consiste de un vector de entrada x_t y una salida deseada z_t. Para la tarea de clasificación de z_t es usual usar $+1$ o -1. La regla de aprendizaje del perceptrón es relativamente simple y la describimos de la siguiente manera:

i) Comenzar con pesos aleatorios para las conexiones,

ii) Seleccionar el vector x_t para el entrenamiento,

iii) Si $|\overline{z}_t - z_t| > 0$, modificar todas las conexiones $w(i)$ de acuerdo a: $z_t(i)x_t(i)$

iv) Regresar al paso 2.

1.9.2. Teorema de convergencia del perceptrón

Para la regla de aprendizaje del perceptrón existe un teorema de convergencia, el cual estable lo siguiente

Teorema 1.2 *Si existe un conjunto de pesos* $\mathbf{w}^*$ *el cual es capaz de realizar la transformación* $\overline{z}_t$*, la regla de aprendizaje del perceptrón convergerá a una solución en una cantidad finita de pasos para cualquier elección inicial de los pesos.*

Prueba

El tamaño del vector $\mathbf{w}^*$ no es significativo, así que tomemos $||\mathbf{w}^*|| = 1$. Además, como $\mathbf{w}^*$ es una solución correcta, el valor $|\mathbf{w}^* \cdot \mathbf{x}|$ es mayor que cero. Podría darse el caso que $\mathbf{w}^* \cdot \mathbf{x} = 0$ para un $\mathbf{w}^*$ el cual no produce una clasificación errónea. Sin embargo, podemos encontrar otro $\mathbf{w}^*$ tal que el producto no sea cero.

Como $|\mathbf{w}^* \cdot \mathbf{x}| > 0$ es equivalente a que existe un $\delta > 0$ tal que $|\mathbf{w}^* \cdot \mathbf{x}| > \delta$ para todas las entradas $\mathbf{x}$.

De acuerdo a la regla de aprendizaje del perceptrón, los pesos de las conexiones son modificados para una entrada dada $\mathbf{x}$, y los nuevos pesos es dado por $\mathbf{w}' = \mathbf{w} + \overline{z}\mathbf{x}$. De esto se desprende que

$$\begin{aligned} \mathbf{w}' \cdot \mathbf{w}^* &= \mathbf{w} \cdot \mathbf{w}^* + \overline{\mathbf{z}}\mathbf{x} \cdot \mathbf{w}^* \\ &= \mathbf{w} \cdot \mathbf{w}^* + \operatorname{sgn}(\mathbf{w}^* \cdot \mathbf{x})\mathbf{w}^* \cdot \mathbf{x} \\ &> \mathbf{w} \cdot \mathbf{w}^* + \delta \end{aligned}$$

Por otro lado, tenemos que

$$\begin{aligned} ||\mathbf{w}'||^2 &= ||\mathbf{w} + \bar{z}\mathbf{x}||^2 \\ &= \mathbf{w}\cdot\mathbf{w} + 2\bar{z}\mathbf{w}\cdot\mathbf{x} + \mathbf{x}\cdot\mathbf{x} \\ &< \mathbf{w}\cdot\mathbf{w} + \mathbf{x}\cdot\mathbf{x} \\ &= \mathbf{w}\cdot\mathbf{w} + M \end{aligned}$$

Después de k modificaciones, se tiene que

$$\mathbf{w}_k \cdot \mathbf{w}^* > \mathbf{w}\cdot\mathbf{w}^* + k\delta$$

$$||\mathbf{w}_k||^2 < \mathbf{w}\cdot\mathbf{w} + kM$$

tal que

$$\begin{aligned} \cos(\theta_k) &= \frac{\mathbf{w}^*\cdot\mathbf{w}_k}{||\mathbf{w}^*||||\mathbf{w}_k||} \\ &= \frac{\mathbf{w}^*\cdot\mathbf{w}_k}{||\mathbf{w}_k||} \\ &> \frac{\mathbf{w}^*\cdot\mathbf{w} + k\delta}{\sqrt{\mathbf{w}\cdot\mathbf{w} + kM}} \end{aligned}$$

$$\lim_{k\to\infty} \cos(\theta_k) > \lim_{k\to\infty} \frac{\mathbf{w}^*\cdot\mathbf{w} + k\delta}{\sqrt{\mathbf{w}\cdot\mathbf{w} + kM}} = \infty$$

mientras que por definición $-1 \leq \cos(\theta) \leq 1$.

Por lo tanto, existe una cota superior $k_{\text{máx}}$ para k. Es decir, después de $k_{\text{máx}}$ las variaciones de los pesos se está realizando correctamente. $k_{\text{máx}}$ es alcanzada cuando $\cos(\theta) = 1$. Si iniciamos con $\mathbf{w} = \mathbf{0}$,

$$k_{\text{máx}} = \frac{M}{\delta^2}$$

■

Algorithm 1 Algoritmo del Perceptrón

procedure PERCEPTRON($\mathbf{w_0}$)
  $\mathbf{w_1} \leftarrow \mathbf{w_0}$ ▷ Usualmente $\mathbf{w_0} = \mathbf{0}$
  for $t \leftarrow 1, T$ **do**
    RECIBIR($\mathbf{x_t}$)
    $\overline{z}_t \leftarrow \text{sgn}(\mathbf{w_t} \cdot \mathbf{x_t})$
    RECIBIR(z_t)
    $error = |\overline{z}_t - z_t|$
    if $error > 0$ **then**
      $\mathbf{w_{t+1}} \leftarrow \mathbf{w_t} + z_t \mathbf{x_t}$
    else
      $\mathbf{w_{t+1}} \leftarrow \mathbf{w_t}$
    end if
  end for
  return $\mathbf{w_{T+1}}$;
end procedure

1.10. El algoritmo de propagación de errores

Minsky y Papert mostraron en 1969 que una red de retroalimentación con dos capas puede tener muchas restricciones, pero no presentaron una solución del problema de como ajustar los pesos de las entradas de la capa oculta.

El término *backpropagation* (propagación hacía atrás) fue acuñado por Frank Rosenblatt y una respuesta de cómo resolver el ajuste de los pesos fue presentado por David Rumelhart, Geoffrey Hinton y Ronald Williams en 1986, y soluciones similares aparecieron publicadas posteriormente.

La idea central detrás de esta solución es que los errores de las unidades de la capa oculta son determinados por la propagación hacia atrás de los errores de la capa de salida. Por esta razón el método es a veces llamado *backpropagation learning*

rule. Backpropagation, considerado como una generalización de la regla delta para funciones de activación no lineal y redes multicapas.

1.10.1. La regla delta

Si pretendemos minimizar una función de error, podríamos usar el error cuadrático medio de la forma

$$E_{MSE} = \frac{1}{2}\frac{1}{n}\sum_j (t_j - z_j)^2$$

o podemos tomar la suma de los residuos al cuadrado dada por

$$E_{SSE} = \frac{1}{2}\sum_j (t_j - z_j)^2$$

tomando el j-ésimo elemento,

$$E_j = \frac{1}{2}(t_j - z_j)^2$$

nuestra medida de error sobre el entrenamiento la podemos reescribir como

$$E = E_{SSE} = \sum_j E_j$$

Consideremos el caso unidimensional, para minimizar el error, podemos calcular la primera derivada (gradiente en el caso vectorial) y realizar una corrección en los pesos en el sentido opuesto, como se muestra en la gráfica.

La derivada será positiva cuando se encuentre en el lado derecho de la parábola y como el objetivo es minimizar ese error, debemos disminuir el peso actual w para acercaros al valor óptimo w^*.

Entonces consideremos neuronas lineales, con salida z, que es una combinación lineal de los pesos w y sus entradas x dada en la expresión.

$$z = w^t x = w \cdot x = \sum_i w_i x_i$$

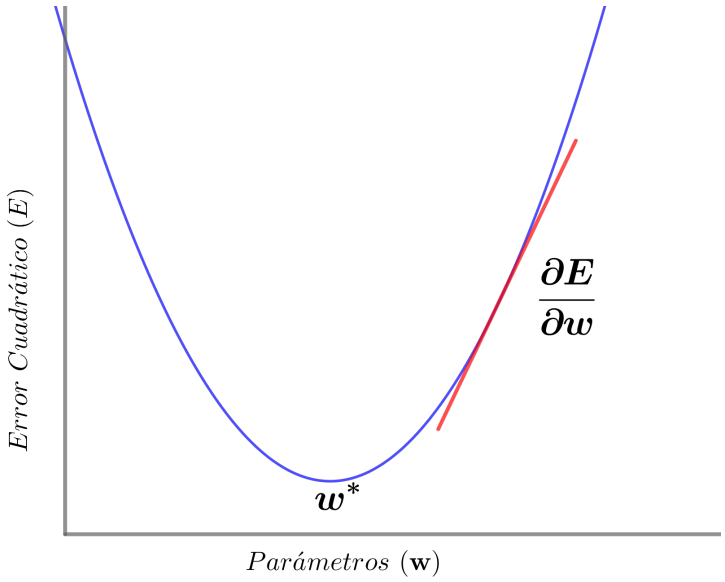


Figura 1.13: Dirección de la primera derivada de E

La derivada parcial E con respecto a w_i viene dada por

$$\frac{\partial E}{\partial w_i} = \sum_j \frac{\partial E_j}{\partial w_i} = \sum_j \frac{\partial E_j}{\partial z} \frac{\partial z}{\partial w_i}$$

de esta manera hemos utilizado la definición del error y la regla de la cadena para calcular la derivada parcial del error con respecto a los pesos, como un producto de dos derivadas parciales que resulta ser la del error con respecto a la salida y la de la salida con respecto a los pesos.

Calculando estas dos derivadas parciales obtenemos

$$\frac{\partial E}{\partial z} = -(t_j - z_j)$$

y

$$\frac{\partial z}{\partial w_i} = x_i$$

reemplazando en la derivada parcial del error con respecto a los pesos obtemos

$$\frac{\partial E}{\partial w_i} = -\sum_j (t_j - z_j) x_{ji}$$

donde x_{ji} representa el valor de la entrada i para el patrón de entrenamiento j.

Consideremos el operador gradiente definido por

$$\nabla = \left(\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \cdots, \frac{\partial}{\partial w_n} \right)$$

de manera que las derivadas parciales del error de nuestra neurona lineal se pueden escribir como

$$\nabla E = -\sum_j (t_j - z_j) x_j$$

y la cantidad que usamos para ajustar nuestro peso es

$$\Delta w = -\eta \nabla E = \eta \sum_j (t_j - z_j) x_j$$

donde η se llama tasa de aprendizaje.

Ejemplo **1.8** *Supongamos que tenemos un contrato con una telefónica que utiliza una tarifa según la cantidad de minutos de la llamada por ejemplo: hasta 5 minutos, 10 minutos y 30 minutos.*

El importe a nuestra cuenta es determinado por la siguiente relación lineal

$$total = w_{5m} x_{5m} + w_{10m} x_{10m} + w_{30m} x_{30m}$$

donde w_i corresponde al precio individual asociado a la cantidad de minutos y x_i es el número de unidades consumidas.

Iniciaremos a partir de una estimación «aleatoria», de los precios a cada consumo.

$$w_{5m} = 3$$

$$w_{10m} = 3$$

$$w_{30m} = 2$$

Estamos asumiendo que desconocemos los precios reales, que son los siguientes:

$$w^*_{5m} = 4$$

$$w^*_{10m} = 2$$

$$w^*_{30m} = 1$$

Supongamos que en un mes consumimos 4 paquetes de 5 minutos, 6 de 10 minutos y 12 de 30 minutos, la factura que nos debería de llegar al final del mes es

$$\begin{aligned} factura &= \sum_i w^*_i x_i \\ &= 40 \end{aligned}$$

Con nuestra estimación esperábamos recibir

$$\begin{aligned} estimación &= \sum_i w_i x_i \\ &= 54 \end{aligned}$$

Por lo que hemos cometido un error de estimación de $+14$*:*

$$error = estimación - factura = +14$$

Utilizaremos la regla delta para actualizar los precios estimados, que modifica los precios (pesos) en dirección opuesta al error cometido, con el objetivo de reducir el error. Es decir,

$$\Delta w_i = -\eta \; error \; x_i = \eta(factura - estimación)$$

donde η es la tasa de aprendizaje utilizada para controlar el tamaño de los ajustes, escoger este valor resulta crucial. Supongamos que para nuestro caso $\eta = 1/140$:

$$\Delta w = (-0.4, -0.6, -1.2)$$

Nuestra nueva estimación de los precios pasará a ser

$$w(1) = w(0) + \Delta w = (2.6, 2.4, 0.8)$$

Realizamos de nuevo nuestra estimación de la factura obteniendo

$$\begin{aligned} estimación &= \sum_i w_i x_i \\ &= 34.4 \end{aligned}$$

Observemos que el error se ha reducido, en términos absolutos, puesto que el error es de -5.6 y el ajuste de los pesos fue demasiado drástico. Esto se debe a que el precio real del paquete de 5 minutos es de 4 y no 3 como se pensaba, actualizándose a 2.6. El error en general se reduce, ya que los precios de los otros paquetes se ajustaron lo suficiente.

Es posble afinar más los resulados, si repetimos el proceso de ajustar los pesos varias veces usando las facturas de meses posteriores.

1.10.2. La regla delta generalizada

Consideremos una función de activación f diferenciable, la entrada total se define como

$$y_k^p = f(z_k^p)$$

donde

$$z_k^p = \sum_j w_{jk} y_j^p + \theta_k \tag{1.4}$$

La generalización de la regla delta está dado por

$$\Delta_p w_{jk} = -\eta \frac{\partial E^p}{\partial w_{jk}}$$

y la suma de los residuos al cuadrado es

$$E^p = \frac{1}{2} \sum_{i=1}^{N_0} (t_i^p - y_i^p)^2$$

donde t_i^p es la salida deseada para la unidad i cuando el patrón p es fijo.

Nuestra medida de error esta dado por

$$E = \sum_p E^p$$

y la podemos escribir como

$$\frac{\partial E^p}{\partial w_{jk}} = \frac{\partial E^p}{\partial z_k^p} \frac{\partial z_k^p}{\partial w_{jk}}$$

Por la ecuación (1.4) el factor el segundo factor es

$$\frac{\partial z_k^p}{\partial w_{jk}} = y_j^p$$

Si definimos

$$\delta_k^p = -\frac{\partial E^p}{\partial z_k^p}$$

obtenemos una regla delta equivalente descrita en el punto 1.10.1.

$$\Delta w_{jk} = \eta \delta_k^p y_j^p$$

Capítulo 2

El Modelo

2.1. El modelo basado en autómatas celulares

Para un sistema complejo que involucre ecuaciones diferenciales, es de utilidad modelar su comportamiento de forma más simple por medio de los autómatas celulares.

El modelo, con autómata celulares, nos permite experimentar y observar algunas características del sistema en un menor tiempo, además de estudiar las relaciones macro y micro del mismo, lo cual sería imposible con otros modelos. Por ejemplo, consideremos cada célula como un individuo autónomo que toma sus propias decisiones basado en condiciones del entorno y su estado actual, y el efecto que tiene estas decisiones individuales en el desarrollo del fenómeno en general.

Podemos definir el autómata celular como el par (T, f) donde:

- La topología T es un cuadrado (las células en el exterior siempre tienen un mismo estado) y la vecindad es la de Moore, revisado en el capítulo 1.
- f es una función de transición unidireccional para la celda x_{ij} de manera que para ir del estado $S^t_{x_{ij}}$ al estado $S^{t+1}_{x_{ij}}$ dependerá del estado actual de la célula

$S^t_{x_{ij}}$ y del conjunto de estados de las células en la vecindad $S^t_{\mathcal{N}(i,j)}$, definida de la forma siguiente

$$f(S^t_{x_{ij}}, S^t_{\mathcal{N}(i,j)}) = \begin{cases} 1 & , S^t_{x_{ij}} = 0 \text{ y } \#(S^t_{\mathcal{N}(i,j)}) \geq 1 \\ 2 & , S^t_{x_{ij}} = 1 \text{ y } PeriodoLatente \geq 1 \\ 3 & , S^t_{x_{ij}} = 2 \text{ y } TiempoRecuperación \geq 5 \\ 3 & , S^t_{x_{ij}} = 3 \text{ y } TiempoInmunidad \geq 7 \\ S^t_{x_{ij}} & , \text{en otro caso.} \end{cases}$$

donde $S = \{0, 1, 2, 3\}$ es el conjunto de estado con $S \to 0$, $E \to 1$, $I \to 2$, $R \to 3$ y $\#(\cdot)$ es la función que devuelve la cantidad de estados igual a 2 como se muestra en la figura siguiente.

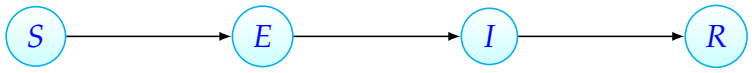


Figura 2.1: Transición de los estados del modelo AC tipo SEIR

Los estados por lo que puede pasar una persona, Se describe de la siguiente manera:

- $S \to$ **Susceptible:** Persona es susceptible de contraer la enfermedad.
- $E \to$ **Infectado:** Individuo que ha contraído la enfermedad, pero no tiene síntomas ni puede contagiar, se encuentra en el período de incubación y después pasará al estado I.
- $I \to$ **Infeccioso:** Persona que empieza a desarrollar síntomas y puede contagiar a otras personas, el período esperado en este estado puede ser cambiado debido a las medidas de control.
- $R \to$ **Recuperado:** Individuo que ha sobrevivido al virus, no es contagioso y no puede contraer la enfermedad de nuevo pues ha desarrollado inmunidad natural.

Nuestro estudio se centra en la ciudad de Panamá, de manera que en el caso de

una epidemia es el Ministerio de Salud de Panamá y otras entidades las encargadas de activar protocolos de seguridad. Ponen en marcha medidas de control con el objetivo de vigilar y reducir el impacto de una pandemia. Debido, a esto, es importante considerar algunas de estas medidas en el modelo, que la designaremos modelo versión 2. Las medidas consideradas son:

Confinamiento: Las personas infectadas son aisladas del resto de la población para evitar contagios.

Cuarentena: Se realizan controles en la zona de contagio y se restringen los movimientos de posibles personas infectadas.

2.2. Propagación dinámica

Para modelizar la propagación dentro de un área concreta, se utilizará un modelo estocástico, con las siguientes suposiciones:

- Existe una pequeña cantidad de individuos infecciosos, lo denotaremos como I_0.
- Los individuos latentes no son infecciosos.

Además, en lugar de considerar tasas de transición entre estados (Brauer y col., 2008), se definen intervalos de tiempo (en días) que va a depender del individuo y la evolución de la epidemia que se define en el algoritmo de autómata celular (Algorithm 2)

Algorithm 2 Algoritmo de autómata celular

```
1:  procedure AC(M_0, T)
2:      n ← len(M_0(1, :))
3:      Malla ← M_{n×n}
4:      Malla ← M_0
5:      for t ← 1, T do
6:          for i ← 1, n do
7:              for j ← 1, n do
8:                  EXPUESTO (i, j)
9:                  INFECTADO (i, j)
10:                 RECUPERADO (i, j)
11:             end for
12:         end for
13:     end for
14:     return Malla;
15: end procedure
```

Los distintos grupos S, E, I y R en el instante de tiempo t, quedan definidos por los siguientes algoritmos: Estado expuesto (Algorithm 3), Estado infectados (Algorithm 4) y Estado recuperados (Algorithm 5).

Algorithm 3 Estado expuesto

```
1: procedure EXPUESTO(i, j)
2:     estado ← 0
3:     if MALLA(i, j) = 0 then
4:         for l ← i − 1, i + 1 do
5:             for l ← j − 1, j + 1 do
6:                 if MALLA(k, t) = 2 then estados ← estados + 1
7:                 end if
8:             end for
9:         end for
10:    end if
11:    if estados ≥ 1 then Malla(i, j) ← 1
12:    end if
13:    return Malla(i, j);
14: end procedure
```

Algorithm 4 Estado infectados

```
1: procedure INFECTADO(i, j)
2:     periodolatente ← randint(0, 5)
3:     if Malla(i, j) = 1 and periodolatente < randint(1, 2) then Malla(i, j) ← 2
4:     end if
5:     return Malla(i, j);
6: end procedure
```

Algorithm 5 Estado recuperados

1: **procedure** RECUPERADO(i, j)
2: **tiemporecuperacion** $\leftarrow$ **randint**$(\mathbf{0}, \mathbf{15})$
3: **tiempoinmunidad** $\leftarrow$ **randint**$(\mathbf{0}, \mathbf{15})$
4: **if** Malla$(i, j) = 2$ **and** $tiemporecuperacion \geq$ **randint**$(\mathbf{1}, \mathbf{7})$ **then** Malla$(i, j) \leftarrow 3$
5: **end if**
6: **if** Malla$(i, j) = 3$ **and** $tiempoinmunidad \geq 7$ **then** Malla$(i, j) \leftarrow 3$
7: **end if**
8: **return** Malla(i, j);
9: **end procedure**

2.3. Salida del modelo AC

Consideremos otros parámetros finales de la simulación como:

$acumulado_{infectados}(T)$ que devuelve el número acumulado de infectados en el instante T y que se determina de la siguiente forma:

$$acumulado_{infectados}(T) = acumulado_{infectados}(0) + \sum_{t=1}^{T} I(Malla, t)$$

donde $I(Malla, t)$ es el número de infectados en la simulación en el instante t.

De forma análoga se puede definir $acumulado_{susceptibles}(T)$, $acumulado_{Expuestos}(T)$ y
$acumulado_{Recuperados}(T)$.

El parámetro $acumulado_{infectados}(T)$ juega un papel importante para determinar la dimensión de la epidemia y uno de los indicadores que utiliza el Ministerio de Salud de Panamá (MINSA) y la Organización Mundial de la Salud (WHO).

2.4. Predicción con redes neuronales

Las variables que consideraremos, para el estudio, serán las mismas utilizadas para pronosticar cuando un paciente contrae COVID-19, entre las cuales tenemos: la fatiga, expectoración, nariz tapada, fiebre, tos seca, entre otras (Michelen y col., 2020).

Por lo tanto, para predecir la severidad de la enfermedad, es razonable considerar los síntomas como datos de entrada. En este estudio se utilizará, para la prueba, el conjunto de datos del COVID-19 *Symptoms Checker* que consta de 31680 registros para todas las variables. Utilizaremos el 70 % para el modelo de predicción y el 30 % restante para verificar el modelo.

El modelo RN es desarrollado para predecir cuales, de las personas, infectadas, desarrollaran una situación de gravedad de la enfermedad.

Es importante destacar que el modelo utilizado es una RN multicapa de manera que se desarrollen las combinaciones no lineales de las variables de entrada. Se empleará funciones de activación RELU y Softmax en las capas ocultas, y en el nodo de salida se utiliza también RELU, revisado en el capítulo 1.

Las RN cuentan con seis capas ocultas (ver Figura 2.2) y se entrena utilizando un algoritmo de gradiente descendiente estocástico (SGD) y términos de momento. El número de nodos varía entre 5 y 20 en la que en cada iteración se calcula el error cuadrático medio (MSE) entre la salida del modelo y los datos.

Para fines de predicción, la propiedad más importante de un modelo es su competencia para generalizar. Si bien la competencia de generalización indica el poder de un modelo en el buen desempeño con datos no utilizados para entrenarlo. El sobreajuste evita la generalización del modelo frente a nuevas situaciones (Bishop, 1995), así para evitar esto, se emplea la interrupción temprana (*EarlyStopping*) de la técnica de regularización más utilizada y con el uso de la capa de abandono (*Dropout*).

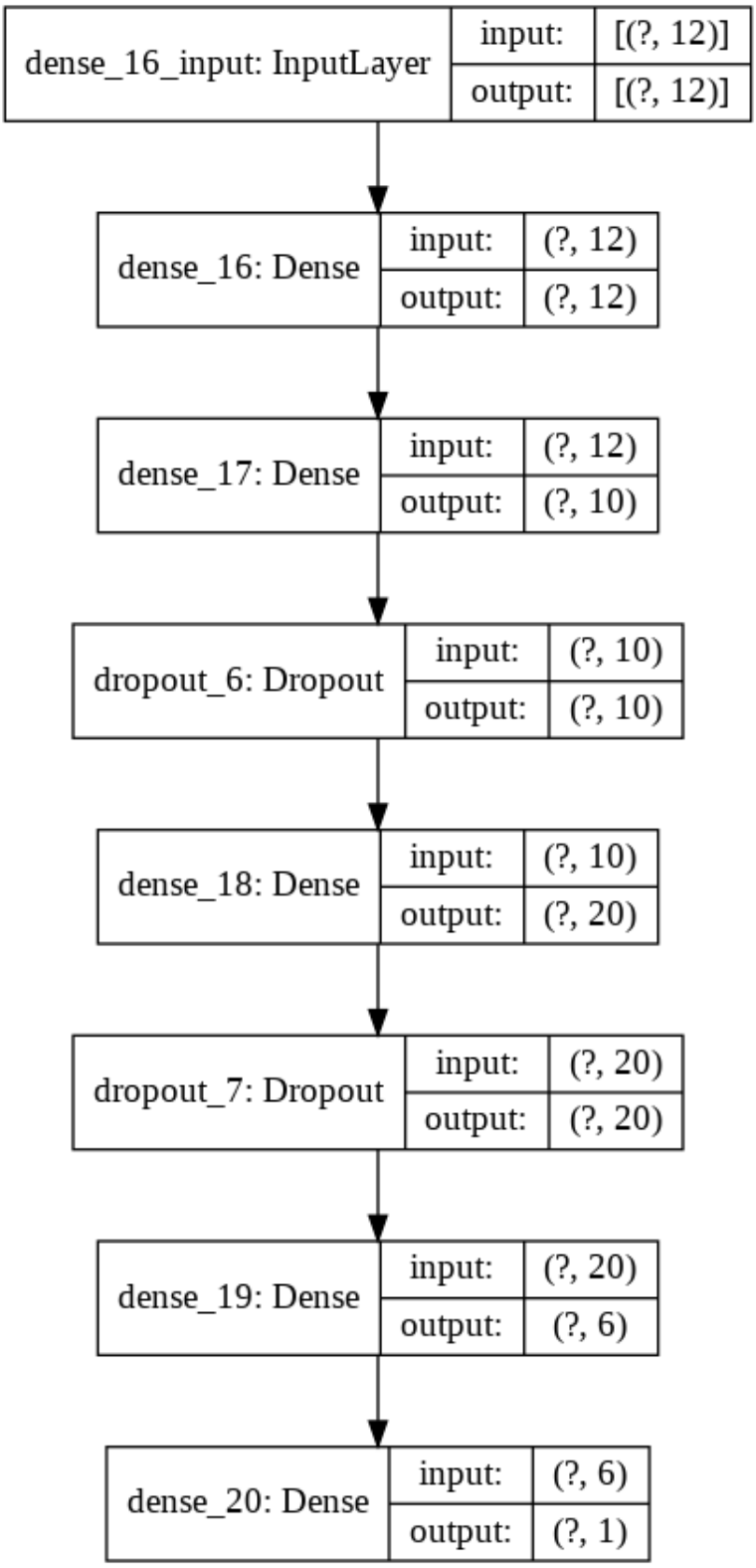


Figura 2.2: Gráfica del modelo de red neuronal

Capítulo 3

Experimentos y resultados

3.1. Diseño de los experimentos

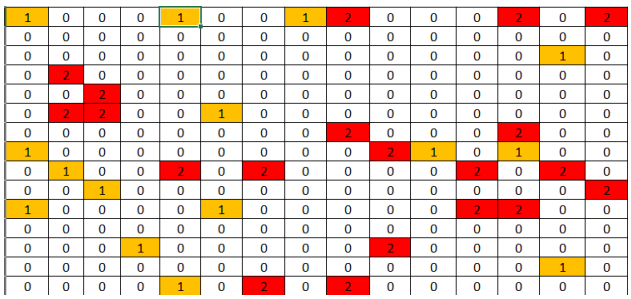
Una de las limitaciones, ya mencionada, para un estudio de propagación de enfemedades es contar con una data que contenga información oportuna y válida. En el caso que nos compete fue imposible encontrar una base de datos para los casos de Influenza, en Panamá, acorde con las necesidades del modelo en estudio. Es por ello que al revisar algunas investigaciones y publicaciones de las entidades encargadas del estudio del compotamiento de enfermedades, se decidió aplicar los datos sobre el comportamiento de la influenza proporcionado por la (CDC, 2019) de 225 personas analizadas, asumiendo que ese comportamiento del virus en otros países, es similar en Panamá.

Para la implementación se estableció los parámetros que alimentaran la simulación y las propiedades de los autómatas celulares los cuales muestran en el cuadro 3.1:

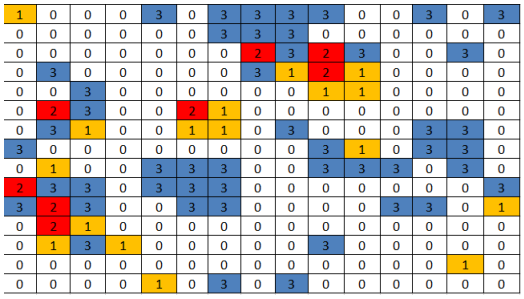
Parámetro	Significado	Unidad de medida
N	Número de personas	Personas
T	Tiempo total de la simulación	Días
S_t	Número de susceptibles al inicio de la temporada	Personas
E_t	Número de expuestos al inicio de la temporada	Personas
I_t	Número de infectados al inicio de la temporada	Personas
R_t	Número de recuperados al inicio de la temporada	Personas
η_1	Tiempo de incubación de la influenza	Días
η_2	Tiempo de recuperación de la influenza	Días
η_1	Tiempo de inmunidad de la influenza	Días

Cuadro 3.1: Parámetros para la implementación del modelo

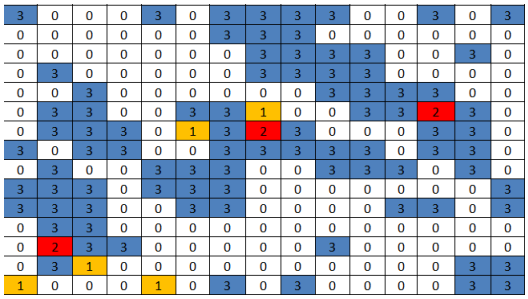
La evolución del virus en la población aparece en la figura 3.1 , en que sólo se muestra una porción de la población en un instante de tiempo, para posteriormente observar y conocer en que intervalo de un tiempo toda la población es infectada.



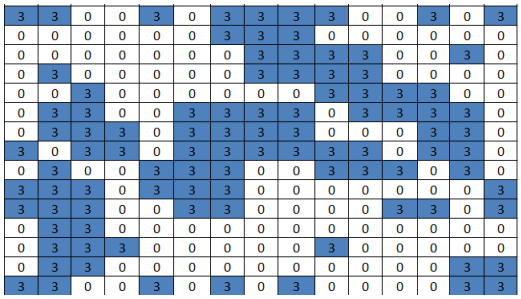
(a) Modelo en $t = 0$



(b) Modelo en $t = 2$



(c) Modelo en $t = 6$



(d) Modelo en $t = 30$

Figura 3.1: Un mapa que muestra el estado de la epidemia para $t = 0,\ 2,\ 6,\ 30$.

La entrada del modelo es una matriz compuesta por la población de susceptibles, expuestos, infectados y recuperados, basándose en la vecindad de Moore, originando tres funciones: $EXPUESTOS(\cdot, \dot{)}$, $INFECTADO(\cdot, \cdot)$ y $RECUPERADO(\cdot, \cdot)$ que tienen el trabajo de cambiar el estado de la celda, si se cumple las condiciones.

Además, se toma en forma aleatoria el periodo de incubación, recuperación e inmunidad para comprobar si se cumple tanto para el tiempo de cada estado por cada persona.

La salida del programa es entregar los vectores de los susceptibles, expuestos, infectados y recuperados en cada tiempo t. Para esto se tomaron los siguientes valores:

Parámetro	Significado	Datos de la simulación
N	Número de personas	225
T	Tiempo total de la simulación	30
S_t	Número de susceptibles al inicio de la temporada	190
E_t	Número de expuestos al inicio de la temporada	15
I_t	Número de infectados al inicio de la temporada	20
R_t	Número de recuperados al inicio de la temporada	0
η_1	Tiempo de incubación de la influenza	1-2
η_2	Tiempo de recuperación de la influenza	1-7
η_3	Tiempo de inmunidad de la influenza	7

Cuadro 3.2: Valores de los parámetros de una simulación

Manteniendo las condiciones anteriores, se añade además el confinamiento cuando se detecta una persona expuesta e infectada, considerando los siguientes parámetros

Parámetro	Significado	Unidad de medida
ρ_1	Porcentaje de expuestos detectados	-
ρ_2	Porcentaje de infectados detectados	-
C	Tiempo de cuarentena	Días

Cuadro 3.3: Parámetros para la implementación del modelo versión 2

Parámetro	Significado	Unidad de medida
ρ_1	Porcentaje de expuestos detectados	20 %
ρ_2	Porcentaje de infectados detectados	20 %
C	Tiempo de cuarentena	10

Cuadro 3.4: Valores de los parámetros de una simulación del modelo versión 2

Para el segundo modelo de redes neuronales, descrito en el punto 2.4, es impor-

tante primero hacer un pre procesamiento para verificar la "limpieza"de los datos, es decir que estén completos, hecho que no hizo falta por que la data utilizada está completa.

Se realizaron hasta 1000 iteraciones, en la experimentación de entrenamiento, teniendo en cuenta que se está empleando la interrupción temprana y que las funciones de activación en las capas se mantendrán iguales.

El optimizador utilizado, primero es el SGD luego se cambiará a Adam para comparar los errores de entrenamiento y generalización.

3.2. Resultados

En la figura 3.2 aparecen las curvas solución del modelo dinámico del primer escenario que refleja el comportamiento epidémico de la influenza.

Tomando las tasas de detección de 20 % de expuestos y 30 % de infectados para él segundo escenarios, obtenemos las curvas solución en la figura 3.3 y comparándolas con las anteriores, podemos notar que a partir del día 5 se produce una disminución más pronunciada del número de personas infectadas diariamente, mientras que el primer escenario puede ocurrir un rebrote. La figura 3.4 muestra que el impacto de la epidemia podría ser menor al implementar el confinamiento para las personas expuestas e infectadas.

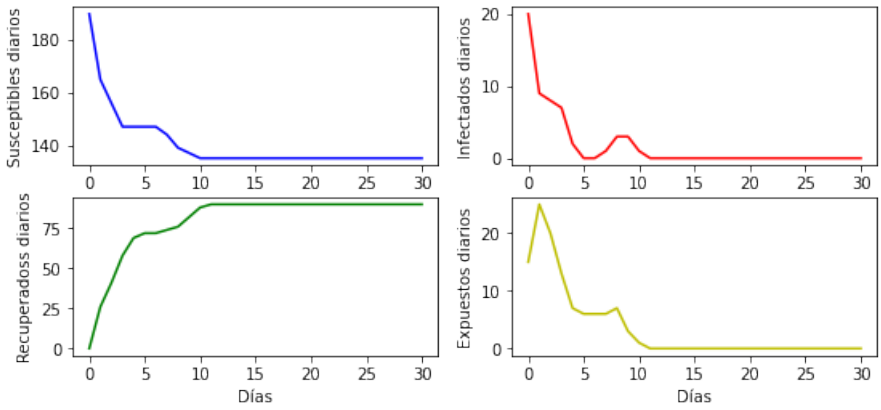


Figura 3.2: Modelo original de autómatas celulares

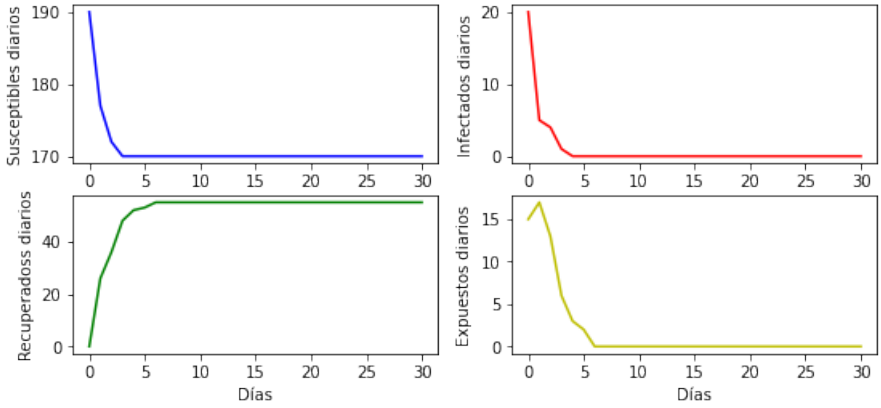


Figura 3.3: Modelo de autómatas celulares con confinamiento

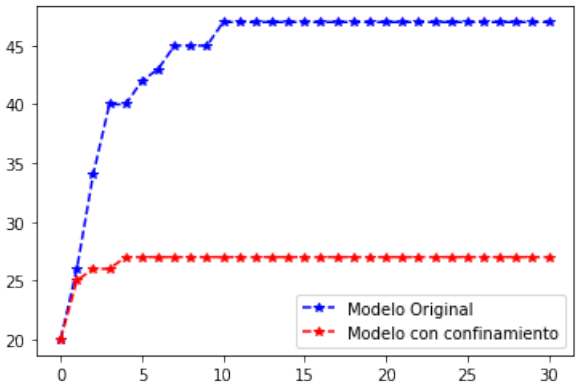


Figura 3.4: Número acumulado de infectados

En la revista «*Pandemic Potential of a Strain of Influenza A (H1N1): Early Finding*» (Fraser y col., 2009) indica que el número reproductivo básico R_0 para la influenza (Cepa pandémica 2009), está entre 1.4 - 1.6 y en la revista «*Modeling influenza epidemics and pandemics: insights into the future of swine flu (H1N1)*» (Coburn y col., 2009) las cepas estacionales están entre 0.9 – 2.1 con promedio 1.5. Para el estudio que desarrollamos, el número reproductivo básico se desterminó utilizando el « Modelo SIRD con ajuste de parámetros». Los parámetros β y γ, que son los necesarios para obtener el R_0 que se define como

$$R_0 = \frac{\beta}{\gamma}$$

En nuestro modelo original (sin confinamiento) la estimación del R_0 es 1.6 lo que muestra que tiene un comportamiento similar a una cepa en una población homogénea y en el caso del modelo con confinamiento y la identificación temprana de los expuestos e infectados el R_0 es 1.4, expuesto en el cuadro 3.5.

	Número de infectados acumulados	Número reproductivo básico (R_0)
Modelo Original	47	1.6
Modelo con confinamiento	27	1.4

Cuadro 3.5: Comparación entre los R_0

Además, en el análisis del modelo de redes neuronales, en las figuras 3.5 y 3.6 correspondiente al entrenamiento del modelo es fácil Observar como el patrón en el entrenamiento tiene un comportamiento parecido a la prueba. Además como la convergencia de ambos coinciden en un valor cercano a la iteración 315 y 68, respectivamente.

Lo anterior nos evidencia que el modelo propuesto tendría una buena estimación al utilizar datos reales.

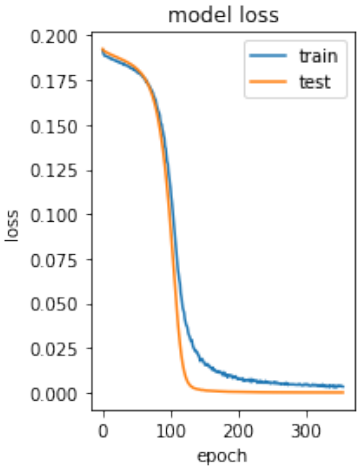


Figura 3.5: Modelo de red neuronal con SGD

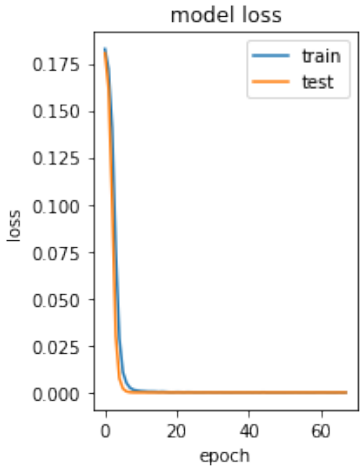


Figura 3.6: Modelo de red neuronal con Adam

Tomando en cuenta que para la evaluación de cada uno de los algoritmos de optimización la data se obtuvo de «*COVID-19 Symptoms Checker*», mostrando la eficiencia de cado uno en el cuadro 3.6.

Optimizador	R^2	MSE	MAE
SGD	0.999778805	$4.190119882e-05$	$3.288483711e-03$
Adam	0.999999913	$1.645122019e-08$	$6.462799178e-05$

Cuadro 3.6: Eficiencia de la red neuronal artificial

Conclusiones y recomendaciones

Conclusiones

Dos modelos de propagación epidémica basados en autómatas celulares se han considerado, primero el modelo homogéneo donde toda la población las reglas son aplicadas en todas partes y, en segundo lugar, el modelo con confinamiento en el que asumimos que se identifica un porcentaje de personas expuestas o infectadas y que estas personas no infectan a otra, cuando es identificada.

La identificación temprana de las personas expuestas e infectadas, junto con el confinamiento reduce en la mayoría de los casos de infectados en las simulaciones, teniendo en cuenta que estamos ante un sistema dinámico cualquier pequeño cambio puede alterar el desarrollo de la epidemia en la población.

El modelo presentado en esta tesis se puede implementar para estudiar y visualizar el comportamiento de la epidemia considerando el tiempo e información sobre la región considerada para el respectivo análisis epidemiológico, ya que se pudo determinar que el número reproductivo básico R_0 de las simulaciones es de 1.6 está dentro de lo esperado para la influenza en una población.

Por otro lado, el objetivo del modelo de RN era identificar de forma temprana las personas que podrían presentar cuadros graves de COVID-19 y con esta información elaborar una estrategia y solicitar los recursos mínimos para atender los pacientes. Debido a la falta de datos y la dificultad de adquisición, este estudio proporciona una RN general para la clasificación de los pacientes. Este estudio fa-

cilitó la producción de pruebas analíticas rápidas y repetidas con la utilización de datos que nos permitieron verificar el rendimiento de los modelos. Se examinaron la red utilizando dos optimizadores el SGD y Adam, con las mismas entradas y salidas.

Se comprobó que el rendimiento de las redes, las dos diseñadas, proporcionan un buen modelo para la clasificación. Específicamente, con el optimizador Adam, producen la mejor bondad de ajuste $(R2 = 0.999999913)$ y el error de predicción más bajo $(MSE = 1.645122019e - 08)$.

No existe una regla para seleccionar el número de capas intermedias y el número de neuronas en las RN. La mejor manera de definir estos números es probando diferentes estructuras de modelos con diferentes números de intermedios hasta lograr un resultado que se ajuste a la realidad.

En el estudio, el desempeño de cada red fue comprobado, y la estructura utilizada fue con el número mínimo de capas, seleccionando neuronas para evitar cualquier redundancia.

Los modelos presentados sirven como complementos para otros modelos basados en ecuaciones diferenciales o análisis estadísticos, no intentan reemplazarlos.

Recomendaciones

Estudiar la dinámica de transmisión de cualquier virus requiere de diversos elementos y modelos muy bien estructurados para que puedan simular un comportamiento real. Este estudio puede ser de apoyo a otros o un punto de partida para su expansión. Es por ello que recomendamos el desarrollo de un estudio de la eficacia con la que se manejan los límites de la matriz celular y añadir límites sociales, que es muy importante. El factor límite, si se maneja correctamente, podría brindar resultados más próximos a la realidad. Tres condiciones de límite importantes como la condición de límite de Dirichlet, la condición de límite de Neumann y la condición de límite de Robin podrían explorarse y compararse entre sí.

Para el modelo de redes neuronales resulta difícil de obtener un resultado real, cuando hace falta información. Pero las redes neuronales (RN) ofrecen un medio para reducir los costos analíticos de la información, reduciendo la cantidad de tiempo dedicado a analizar los datos.

Sería interesante estudiar la cantidad de neuronas por capas, diferentes arquitecturas y otros optimizadores para determinar el mejor resultado.

Bibliografía

Athitan, S., Shukla, V. & Biradar, S. (2014). Dynamic Cellular Automata Based Epidemic Spread Model for Population in Patches with Movement. *Hindawi Publishing Corporation. Journal of Computational Environmental Sciences.* https://www.hindawi.com/journals/jces/2014/518053/

Berzal, F. (2019). *Redes neuronales y Deep learning*. España, Editorial Universidad de Granada.

Bishop, C. (1995). *Neuronal networks for pattern recognition*. Gran Bretaña, Oxford university Press.

Brauer, F., den Driessche, P. V. & Wu, J. (2008). *Mathematical Epidemiology*. Alemania, Springer-Verlang.

CDC. (2019). Datos clave sobre la influenza [[Online; acceso 16-Agosto-2020]]. Centros para le Control y la Prevención de Enfermedades.

Coburn, B., Wagner, B. & Blower, S. (2009). Modeling influenza epidemics and pandemics: insights into the future of swine flu (H1N1). *BMC medicine*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2715422/

Fraser, C., Donnelly, C., Cauchemez, S. & et al. (2009). Pandemic Potential of a Strain of Influenza A (H1N1): Early Findings. *Science*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3735127/

Hernández, J., Ramírez, J. & Ferri, C. (2004). *Introducción a la Míneria de Datos*. España, Pearson Prentice Hall.

Michelen, M., Jones, N. & Stavropoulou, C. (2020). In patients of COVID-19, what are the symptoms and clinical features of mild and moderate cases? *The Centre for Evidence-Based Medicine develops, promotes and disseminates better evidence for healthcare.* https://www.cebm.net/covid-19/in-patients-of-covid-19-what-are-the-symptoms-and-clinical-features-of-mild-and-moderate-case/

Romero, N., Cruz, R. & Wainer, G. (2018). Modelamiento Computacional de la Dinamica de Transmisión de la Varicela mediante Automatas Celulares (Cell-DEVS). *Pesquimats.* https://revistasinvestigacion.unmsm.edu.pe/index.php/matema/article/view/13969

White, H., del Rey, M. & Sánchez, R. (2006). A Model Based on Cellular Automata to Simulate Epidemic Diseases. *Cellular Automata. ACRI 2006. Lecture Notes in Computer Science*. https://link.springer.com/chapter/10.1007%2F11861201_36

Wolfram, S. (1994). *Cellular automata and complexity*. Estados Unidos de América, CRC Press.

Glosario

Adam Es un método de descenso de gradiente estocástico que se basa en la estimación adaptativa de momentos de primer y segundo orden. 47

atractor Es un conjunto de valores numéricos hacia los cuales un sistema tiende a evolucionar, dada una gran variedad de condiciones iniciales en el sistema. 12

atractor extraño Un atractor se llama extraño si tiene una estructura fractal. Este suele ser el caso cuando la dinámica en él es caótica, pero también existen atractores extraños no caóticos. 12

axón El axón, cilindroeje o neurita es una prolongación de las neuronas especializadas en conducir el impulso nervioso desde el cuerpo celular o soma hacia otra célula. 15, 16

ciclo límite Es una trayectoria cerrada en el espacio de fase que tiene la propiedad de que al menos otra trayectoria entra en espiral a medida que el tiempo se acerca al infinito o cuando el tiempo se acerca al infinito negativo. 12

clopen Es un conjunto que es a la vez abierto y cerrado, en un espacio topológico. 13

Dropout Es una técnica de regularización para reducir el sobreajuste en redes neuronales artificiales mediante la prevención de coadaptaciones complejas en

los datos de entrenamiento. 41

EarlyStopping Es una técnica de monitoreo y optimización que se encarga de chequear la pérdida para detener el entrenamiento de la red una vez que esta no se modifica. 41

fenómeno de isotropía Es la característica de algunos fenómenos cuyas propiedades físicas no dependen de la dirección en que son examinados. 7

membrana celular Es una capa o bicapa de fosfolípidos y otras sustancias que delimita toda la célula, dividiendo el medio extracelular del intracelular. 15

perceptrón Es una neurona artificial o unidad básica de inferencia en forma de discriminador lineal, a partir de lo cual se desarrolla un algoritmo capaz de generar un criterio para seleccionar un sub-grupo a partir de un grupo de componentes más grande. 25

período de incubación Es el tiempo comprendido entre la exposición a un Agente biológico, y la aparición de los signos y síntomas por primera vez. 36

SGD Es un método iterativo para optimizar una función objetivo con propiedades de suavidad adecuadas. Puede considerarse como una aproximación estocástica de la optimización del descenso del gradiente, ya que reemplaza el gradiente real por una estimación del mismo. 41

Softmax Es una generalización de la función logística a múltiples dimensiones. Se utiliza en la regresión logística multinomial y a menudo se utiliza como la última función de activación de una red neuronal para normalizar la salida de una red a una distribución de probabilidad sobre las clases de salida predichas, según el axioma de elección de Luce. 41