



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

APLICACIÓN DE UNA RED NEURONAL ARTIFICIAL PARA
PRONOSTICAR EL INPC SUBYACENTE EN MÉXICO

TESIS

PARA OBTENER EL TÍTULO DE:

LICENCIADO EN MATEMÁTICAS APLICADAS

PRESENTA:

VERÓNICA RAMÍREZ MONTERROSA

DIRECTORA DE TESIS:

DRA. LILIA ALEJANDRA FLORES CASTILLO

CODIRECTOR DE TESIS:

DR. MIGUEL FLORES ORTEGA

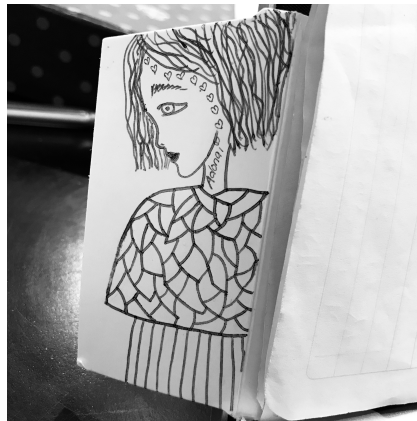
HUAJUAPAN DE LEÓN, OAXACA, MÉXICO

MAYO 2022

Dedicatoria

To GOD.

All the emptyness, the pain, the suffering, the missing sense of my life, that I had in the past 12 continuous years, and during the last days, the killing thoughts, You, God of Abraham, God of Isaac, God of Jacob, You destroyed all that in a breath. I asked for your help, I was dying in pain. You heard me. And You helped me. You didn't only set apart the torture, but You also fulfilled me with so much of Your love and peace.



...and Yeshua is our expected Messiah ♡

Agradecimientos

A DIOS

Por permitirme vivir. Por proporcionarme los medios para poder ayudar en este mundo; y por todas las bendiciones que me ha dado.

A mis padres T. Miguel A. Ramírez Solano y Teresa¹ R. Monterrosa Hernández., por todo su *amor* que siempre me han demostrado durante mi vida. A su vez, a mi hermano, M. Benjamín, por sus consideraciones a mí, en momentos difíciles.

A mi asesora de tesis la Dra. Lilia A. Flores Castillo, por sus contribuciones y comprensiones durante esta etapa; a todos mis profesores de la UTM que contribuyeron de alguna u otra forma a mi formación; y con un distintivo, a dicha formación, para el Dr. Franco Barragán M. A mi codirector de tesis, Dr. Miguel Flores Ortega, por sus comentarios constructivos, y a mis revisores de tesis: Dr. Conrado Aguilar C., Dra. Yannet Paz C., y Dr. Adolfo Maceda M., por su apoyo en la realización de este trabajo. A la UTM, por proporcionarme educación a nivel superior, dentro de un ambiente que enfatiza la naturaleza vegetativa de forma agradable.

A mi papá, por su ayuda incondicional matemática durante la licenciatura. A Henno Brandsma², Keng Hao Ooi³, y Jack⁴, por su ayuda matemática, de forma gratuita, en math.SE⁵; y a Martin Sleziak⁶, por sus pacíficas asistencias en math.meta.SE,⁷ de manera gratuita.

Más agradecimientos y/o agradecimientos de forma explícita, se encuentran en la página⁸: <https://www.overleaf.com/project/62643f580c778008670519bc>

¹Por persuasión de mi mamá, me decidí a iniciar con la realización de esta tesis.

²<https://math.stackexchange.com/users/4280/henno-brandsma>

³<https://math.stackexchange.com/users/284331/user284331>

⁴<https://math.stackexchange.com/users/user9464> (cuenta eliminada)

⁵<https://math.stackexchange.com/>

⁶<https://math.stackexchange.com/users/8297/martin-sleziak>

⁷<https://math.meta.stackexchange.com/>

⁸Se requiere previa solicitud de estos al correo mathmwrk07@gmail.com o un mensaje en https://www.instagram.com/the_release_/

Índice

Índice	i
Glosario	iii
Abreviaturas	iv
Notaciones	v
Introducción	vii
1 Teoría de redes neuronales artificiales	1
1.1. Las redes neuronales artificiales y sus antecedentes	1
1.2. Fundamentos de las redes neuronales artificiales	3
1.3. Tipos de redes neuronales artificiales	5
El perceptrón	7
El perceptrón multicapa	8
1.4. Algoritmo de entrenamiento para redes neuronales multicapa	11
<i>Propagación hacia atrás</i>	11
1.5. Aplicaciones	17
2 La economía, la inflación y su pronóstico	19
2.1. La inflación	19
<i>Inflación subyacente</i>	20
<i>Inflación no subyacente</i>	21
2.2. El Índice Nacional de Precios al Consumidor (INPC)	21
2.3. Metodología de cálculo y construcción del INPC	25
2.4. Efectos de la inflación	29
2.5. Métodos de pronóstico para variables económicas	29
3 Aplicación de la RNA para el pronóstico de la inflación	31
3.1. Datos históricos seleccionados para alimentar la red	31
3.2. Diseño del tipo de red neuronal	32
3.3. Entrenamiento de la red NAR	33
3.4. Selección óptima de la estructura de la red NAR para el primer pronóstico con base en TA	37
3.5. Tasa de la inflación a través del INPC	50
Conclusiones	53

Apéndice	56
Referencias	69

Glosario

INPC: Índice Nacional de Precios al Consumidor. Índice o indicador nacional que permite medir el cambio entre los precios de dos canastas de un tiempo a otro tiempo.

Inflación: Aumento generalizado y continuo de los precios. Se calcula a través de la tasa del INPC.

Inflación subyacente: Aumento generalizado y continuo de los precios no sensibles a 3 tipos de cambios externos: estacionales, gubernamentales y de mercados internacionales. Se calcula a través de la tasa del INPC subyacente.

Especificaciones: Dentro del contexto de la metodología del INPC, son bienes o servicios que poseen de muchas características para describirlos.

Agregado elemental: Es un genérico. También se le llama categoría primaria.

Red tipo autorregresiva no-lineal de Matlab: Red neuronal artificial no lineal autorregresiva capaz de pronosticar datos: $y(t + 1), y(t + 2), \dots$, a partir de recibir n datos previos de la misma serie de tiempo: $y(t - n), y(t - n + 1), \dots, y(t)$.

Perceptrón multicapa: RNA de tipo *feedforward* o de alimentación hacia adelante, es decir, que no se consideran ciclos en la red, tal que puede llegar a aprender a predecir o clasificar información, siempre y cuando se codifique y entrene para ello.

Propagación hacia atrás: Algoritmo de aprendizaje.

Conjunto de entrenamiento: Conjunto de parejas (x_i, y_i) que la RNA considera para poder aprender a simular un comportamiento en específico.

Pronóstico: Modelo que genera nueva información en base a información previa o pasada que se le dio a conocer, a través de uno o más medios en específico.

Abreviaturas

ECM: Error Cuadrático Medio.

INEGI: Instituto Nacional de Estadística y Geografía.

INPC: Índice Nacional de Precios al Consumidor.

RNA: Red Neuronal Artificial.

ENIG: Encuesta Nacional de Ingresos y Gastos de los Hogares.

ENGASTO: Encuesta Nacional de Gastos de los Hogares.

DOF: Diario Oficial de la Federación.

ISO: Organización Internacional para la Estandarización.

SCNM: Sistema de Cuentas Nacionales de México.

CE: Censos Económicos.

CFF: Código Fiscal de la Federación.

IPC: Índice de Precios al Consumidor.

VAR: Vector Autorregresivo.

ARIMA: Movimiento Promedio Integrado Autorregresivo.

RECM: Raíz del Error Cuadrático Medio.

SVM: Máquinas de Vectores de Soporte.

ARM: Análisis de Regresión Múltiple.

NAR: Autorregresiva No-lineal.

ERN: Estructura de la Red Neuronal.

SARS-CoV-2: Síndrome Respiratorio Agudo Severo Coronavirus 2.

Notaciones

- $\mathbb{R}^n \iff$ Espacio vectorial de coordenadas reales, de dimensión n .
- $x \in X \iff x$ pertenece o es un elemento del conjunto X .
- $X^T \iff$ Transpuesta de la matriz X .
- $o^v \iff$ Vector de salida de la capa v de la red neuronal artificial.
- $W^v \iff$ Matriz de pesos que corresponde a la capa número v .
- $b^v \iff$ Vector de bias correspondiente a la capa número v .
- $x_i^v \iff$ i -ésimo elemento del vector x , de la capa v .
- $w_{i,j}^v \iff$ ij -ésimo elemento de la matriz w , de la capa v .
- $\sum_i^m \iff$ Operador que ejecuta la operación $\sum_i^m W_{i,j}^m z_i^m + b_i^m$, donde z_i^m va a representar a las entradas ε_i^m o bien a las salidas o_i^m , de las neuronas.
- $\frac{\partial F^v}{\partial G^d} \iff$ La parcial de la función F en la capa v , respecto a la matriz G de la capa d .

Introducción

A través de la revisión de la historia económica de México y el mundo, un problema presente es la variación en el precio de los productos, a éste comportamiento los economistas lo denominan inflación y se ha realizado un esfuerzo muy grande para entenderlo y controlarlo, la teoría monetaria se encarga de establecer las acciones necesarias para su control, uno de los objetivos importantes de la política económica es promover el crecimiento económico en términos reales que significa que el crecimiento supera a la inflación.

¿Por qué es un problema la inflación? Al definir la inflación como el incremento sostenido en el nivel general de los precios en una economía, el principal problema que se genera es que disminuye el poder de compra de los agentes económicos. ¿Por qué es relevante el estudio de la inflación? Para todo individuo, gobierno o negocio, es importante y de gran interés contar con una herramienta que permita pronosticar el comportamiento de los precios para evitar costos económicos y sociales elevados asociados con periodos de alta inflación y volátil, los cuales tienen repercusiones adversas y significativas sobre el bienestar de la población.

La variación porcentual del Índice Nacional de Precios al Consumidor (INPC) [1] o bien, la inflación, es uno de los peores problemas que puede enfrentar una economía [2], siempre y cuando otros factores de la economía (e.g. el salario) se mantengan fijos o en declive [3]. Así, si la inflación cambia de forma aislada⁹, entonces una de sus mayores consecuencias es que hace que las familias, empresas y gobierno pierdan poder adquisitivo.

Un pronóstico sobre cualquier variable se hace con el fin de que los agentes económicos puedan tomar decisiones de manera más acertada en cuanto a qué, cómo, por qué, y para qué. De esta forma se tendrían más posibilidades de poder afrontar las situaciones futuras con mayor certeza minimizando el riesgo.

El objetivo general del presente trabajo de investigación es diseñar una Red Neuronal Artificial (RNA) que permita pronosticar la inflación subyacente¹⁰ en México por medio del INPC con un nivel mínimo de error.

Los objetivos específicos parten de a) construir una serie de tiempo con una muestra histórica compuesta de observaciones con periodicidad mensual del INPC, de enero del 2011 a enero 2021. Dichos datos se obtendrán del Instituto Nacional de Estadística y Geografía (INEGI); b) Elaborar y analizar: características, alcances, límites, ventajas, desventajas, ilustraciones, ejemplificaciones matemáticas, un modelo con la RNA tipo perceptrón multicapa¹¹, para la obtención del pronóstico de la serie de tiempo del INPC; c) Analizar la forma de entrenamiento para la red perceptrón multicapa, de acuerdo a

⁹De forma aislada hace referencia a que otras cuestiones de la económica (por ejemplo el salario) se mantienen fijos.

¹⁰En sí, se pronóstico la inflación subyacente a partir de la fórmula (3.5), con datos del INPC subyacente que se obtuvieron por medio del pronóstico B.

¹¹La razón por la cual se optó por un perceptrón multicapa, en lugar de otro tipo de red, se fundamenta en que en la revisión de la teoría de redes y en la revisión del estado del arte se identificó que al ser una red que puede tomar la forma de una red no lineal autorregresiva, que puede pronosticar series de tiempo, y mediante la ejecución del algoritmo de propagación hacia atrás puede aproximar cualquier función matemática a cualquier grado de exactitud [4], con mayor rapidez, este tipo de red ha reportado pronósticos con un mínimo de error, siendo esta la principal ventaja de este tipo de red para su aplicación en el pronóstico de cualquier variable.

sus propias características, con el algoritmo *propagación hacia atrás*; d) Implementar la red perceptrón multicapa y entrenarla con el algoritmo *propagación hacia atrás* en el software *Matlab*; e) Evaluar el comportamiento, analizar cuantitativamente y cualitativamente el pronóstico del INPC; f) Determinar la eficiencia del modelo mediante el error cuadrático medio y evaluar el modelo en condiciones de alta inflación.

La hipótesis que guía la presente investigación es la siguiente: Si una red neuronal artificial es una herramienta adecuada para pronosticar la inflación subyacente en México, la cual se representa por medio del INPC, entonces será posible de forma empírica, con la información seleccionada para probar los resultados del pronóstico, medir el error del pronóstico y comprobar que se encuentra dentro del intervalo de $\pm 1\%$.

En esta tesis se considerará un tipo de investigación descriptiva y analítica, con un enfoque cuantitativo, con diseño longitudinal y retrospectivo [5].

En esta investigación se interrelaciona el área de economía, matemáticas e inteligencia artificial, para realizar un pronóstico sobre una variable macroeconómica que es el INPC, mediante la aplicación de una RNA de tipo perceptrón multicapa. La manera de iniciar la investigación es primeramente mediante la revisión de la teoría correspondiente a las redes neuronales artificiales, y con énfasis en el perceptrón multicapa, luego la teoría correspondiente a macroeconomía: INPC e inflación. Y después realizar la aplicación: desde la construcción de la red, su implementación de código en *Matlab*, introducción de los datos del INPC procedentes del INEGI en la red, entrenamiento de la red con el algoritmo de *propagación hacia atrás*, hasta obtener una RNA que tenga la capacidad de llevar a cabo funciones de manera acertada. Con esto, se podrá obtener la predicción de la red sobre el INPC a medio plazo.

CAPÍTULO 1

Teoría de redes neuronales artificiales

1.1. Las redes neuronales artificiales y sus antecedentes

De acuerdo con Matich (2001) en 1936, Alan Turing¹ estudió la mente humana con el fin de establecer una conexión con el área de la computación [7]. Es así que este hecho representa el nacimiento de los primeros indicios sobre la existencia de las redes neuronales artificiales [8]. La investigación formal sobre las redes neuronales siguió avanzando con el neurofísico Warren McCulloch y el matemático Walter Pitts, y en 1943 establecieron las bases teóricas de las redes neuronales artificiales. Ellos hicieron un trabajo que fue publicado con el título “Un Cálculo Lógico de la Inminente Idea de la Actividad Nerviosa”. Probaron que las redes neuronales tenían la capacidad de evaluar cualquier función aritmética o lógica [7, 4]. Además, implementaron el primer modelo neuronal mediante circuitos eléctricos, utilizando una red neuronal artificial simple [7, 8]. Seis años después, Donald Hebb² construyó reglas de aprendizaje para neuronas vivas, esto es, reglas de enseñanza para neuronas desde un punto de vista psicológico [4, 7]. En el año de 1956, se llevó a cabo un congreso en Dartmouth, Canadá, en el cual se dice se vio nacer a la Inteligencia Artificial (IA) [7].

Frank Rosenblatt, en 1957, publicó su trabajo sobre un nuevo concepto, dentro de las redes neuronales, llamado perceptrón [7, 10]. Dentro de las habilidades del perceptrón se encontró que podía generalizar patrones y dentro de las limitaciones de éste se encontró que no podía resolver la función lógica OR-exclusiva, ni tampoco tenía la capacidad de resolver problemas linealmente inseparables [7]. Sin embargo, dicho trabajo fue el más desarrollado hasta esa época [10]. Además, al año del 2001, éste tipo de red perceptrón aún seguía en curso, esto es, 44 años después de su invención [7].

En 1959, Frank Rosenblatt publicó su libro: Principios de Neurodinámica, en el cual se encuentra un teorema que establece las condiciones requeridas para poder tener convergencia en el aprendizaje con una red tipo perceptrón [7].

Tiempo después, Bernard Widroff y Marcian Hoff construyeron una nueva red neuronal llamada *adaptive linear neural network*, que aún al año de 2014 seguía en uso. La red tenía características similares a la red tipo perceptrón, con relación al

¹Matemático, lógico, informático teórico, criptógrafo, filósofo, biólogo teórico, maratoniano y corredor de ultradistancia británico [6].

²Psicólogo, neurocientífico, catedrático y médico [9].

potencial y al diseño. A su vez, introdujeron su algoritmo de la red para entrenarla [4].

En 1961, Karl Steinbeck creó la RNA *Die Lernmatrix*. Esta red introdujo la capacidad de aplicar la memoria asociativa a nivel elemental [7].

En el año de 1969, Marvin Minsky y Seymour Papert publicaron un trabajo en el cual demostraron matemáticamente que el perceptrón era una herramienta de poco rango. Ellos probaron que el perceptrón solo podía aprender funciones lineales. Debido a la magnitud de esta limitación del perceptrón y las repercusiones que tendría en la vida real, el área de las redes neuronales se vio devastada entre la comunidad científica [7].

Paul Werbos, propuso los primeros fundamentos del algoritmo de enseñanza de propagación hacia atrás: *backpropagation* [7], y se considera una valiosa evolución de la teoría de redes neuronales artificiales. Tomó 11 años para que tal algoritmo quedara establecido como teoría [7, 4].

En 1977, Stephen Grossberg creó la Teoría de Resonancia Adaptada (TRA). La “teoría” es en sí una estructura de redes neuronales que, a diferencia de todas las demás estructuras anteriormente creadas, tiene implementada una función que simula la memoria a corto y largo plazo [7].

En 1985, John Hopfield publicó su libro: “Computación neuronal de decisiones en problemas de optimización”. Dada la importancia de los problemas de optimización en el mundo, dicho libro le dio un impulso considerable a las redes neuronales [7].

En 1986, David Rumelhart y G. Hinton trajeron nuevamente a la comunidad científica el algoritmo de *propagación hacia atrás*. Esto fue con el fin de improvisar el algoritmo y por tanto tener mejores resultados en el entrenamiento de las redes neuronales artificiales [7].

Desde los años 80's, las publicaciones y aplicaciones de las redes neuronales artificiales a la vida real siguen vigentes. Es un área que está en constante actividad y agitación tanto para trabajos teóricos como prácticos [4]. De acuerdo a *Google Scholar*³, se ha encontrado una muestra de 20,300 artículos⁴, que hacen referencia a las redes neuronales artificiales en idioma inglés y español. Los artículos forman parte exclusiva del periodo de enero del 2021 al 10 de agosto del 2021.

En la figura 1.1 se presenta un ejemplo de una RNA de tipo multicapa con 3 capas. En este caso particular, se observa una capa de entrada (conformada por las neuronas rojas), una capa oculta (conformada por las neuronas amarillas) y una capa de salida (conformada por las neuronas azules). Las flechas indican la dirección en la que fluye la información de la red, en este caso de izquierda a derecha, denotan las entradas y salidas de las neuronas. Cada salida de una neurona es la entrada de otra neurona, salvo en la última capa. Por último, las flechas a su vez representan la fuerza de conexión que existe entre las neuronas. La manera de mostrar esta relación es mediante los pesos w_x , donde x es cualquier índice representativo de cada neurona y de cada capa.

Los pesos juegan un papel importante en la RNA porque es mediante la

³<https://scholar.google.com/>

⁴Al especificar “Articles in Artificial Neural Networks” en el buscador de *Google Scholar*, con la opción marcada en “Search English and Spanish pages”.

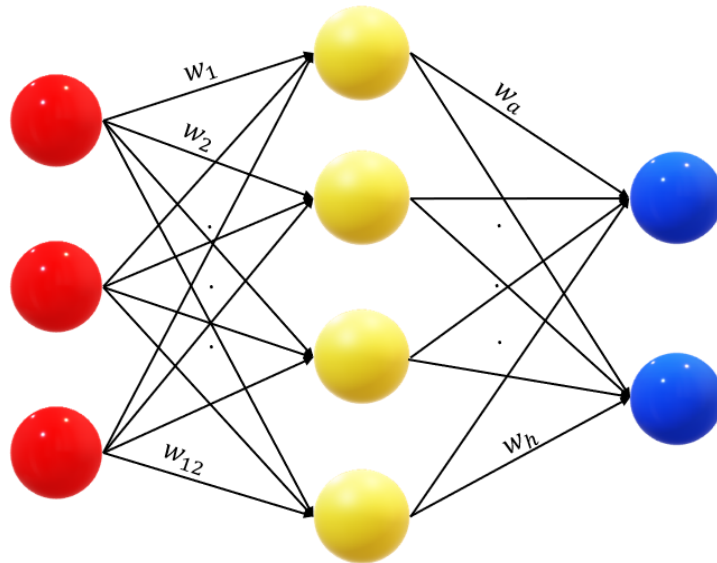


FIGURA 1.1. Ejemplo de una RNA multicapa con 3 capas.
Fuente: Elaboración propia en base a [11].

modificación de ellos por el que la red puede llegar a aprender. Y como consecuencia final, este aprendizaje tiende a dar un buen resultado para la tarea que tenga que ejecutar la red.

Los algoritmos de entrenamiento para las redes neuronales toman lugar en esta parte de aprendizaje de las redes. Son estos los que logran modificar los parámetros de la red para que la red se vuelva autónoma y, en el mejor de los casos, logre el aprendizaje indicado por el diseñador. Si por ejemplo, la salida de la red es incorrecta, el sistema se adapta, a través de un determinado tipo de iteraciones y reglas que involucran la modificación de parámetros, para tener una respuesta correcta. Esta metodología convierte a las redes neuronales en sistemas adaptativos complejos en lugar de solo sistemas complejos, y esto es lo que hace realmente interesante a las redes neuronales artificiales [12].

1.2. Fundamentos de las redes neuronales artificiales

Una neurona, arbitraria, del cerebro humano, puede verse como un objeto conformado por 3 elementos: dendritas (redes de fibras de nervio), cuerpo de la célula y el axón (única fibra extensa). Las dendritas son las partes exteriores de la neurona tales que a través de señales eléctricas se comunican con el cuerpo de la célula. Luego, las señales eléctricas del cuerpo de la célula se filtran por dos medios: restricciones y sumas. De ahí, dichas señales, ya pueden pasar al axón de la neurona, el cual es la salida de la neurona en cuestión y a su vez, es la entrada a otra neurona, nuevamente mediante sus dendritas [4]. Este proceso se lleva a cabo con el fin de que

la información, en el lenguaje de las neuronas (señales eléctricas), se transmite entre la comunidad neuronal. [4] Esta forma de pasar información de una neurona a otra se llama sinapsis. Para terminar, todas las funciones de las neuronas son almacenadas en las mismas neuronas y en las conexiones que existen entre ellas.

El cerebro humano está conformado por un conjunto finito de neuronas, las cuales están relacionadas de manera específica para llevar a cabo funciones [4]. Este sistema de neuronas puede verse como una gran red neuronal humana. [4] La función de la red neuronal humana queda determinada por 3 factores: la fuerza de conectividad que existe entre las neuronas, la manera en la cual dichas neuronas están almacenadas o jerarquizadas y desarrollos químicos avanzados. Cuando nacen conexiones entre las neuronas o cuando se modifican conexiones ya existentes entre ellas, se observa el aprendizaje del ser humano [4]. Esto es, se lleva a cabo una determinada función en la red neuronal humana de tal forma que se genera conocimiento. La neurona consiste de 4 componentes: un operador \sum , un bias b , una función de activación⁵ f y de una salida s . Su función es llevar a cabo una pequeña misión tal que contribuya a la misión general de una RNA [8, 13, 14, 10, 7, 15, 4].

La red consiste de capa(s), en las cuales están almacenadas las neuronas artificiales. Al mismo tiempo, las capas incluyen parámetros denominados pesos w , los cuales proveen de información a las neuronas y además determinan la profundidad de la relación que existe entre dichas neuronas [4]. La función de una RNA es llevar a cabo una misión general que es construida a partir de las pequeñas misiones de cada neurona artificial que la compone. [4] De manera un tanto análoga a una red neuronal biológica, la función de la RNA queda determinada mediante las conexiones entre sus elementos [4].

Dependiendo del tipo de RNA con la que se trabaje, se determinará el número de capas, número de neuronas para cada capa (generalmente cada capa está compuesta por una o más neuronas artificiales), distintas o iguales funciones de activación para cada capa o neurona, el conjunto numérico de pesos, y los bias nulos o activos [8, 13, 14, 10, 7, 15]. También, dependiendo de lo que se quiera obtener de la red, se establecerá el tipo de relaciones que habrá entre las neuronas y la red en general. La forma en que se comporta una RNA, de forma abstracta es como sigue: se introduce información proveniente del exterior y ésta es asimilada por la primer capa de la red o capa de entrada, para después transformar la información y/o crear dentro de ella el o los objetivos deseados para luego compartir los resultados con las capas posteriores. Las capas posteriores a su vez transformarán la información y/o crearán dentro de ellas el o los objetivos deseados, para luego mostrarlo(s) al exterior, a través de la última capa llamada capa de salida.

En sí, la analogía o relación que hay entre una neurona biológica y una neurona artificial, es como sigue [4]: fuerza de conexión de la sinapsis \leftrightarrow peso, cuerpo de la célula \leftrightarrow operador, y función de activación; y señal en el axón \leftrightarrow evaluación o resultado de la función de activación. [10] Una ventaja que se encuentra en las redes neuronales respecto a herramientas comunes es que simula al cerebro humano de manera más eficiente en cuestiones de examinación y captación, para llevar a cabo tareas de cálculo. En la imagen 1.2 se muestra un ejemplo de una neurona “biológica” y en la imagen 1.3 se ve una neurona artificial y sus componentes.

Para concluir la sección, se debe aclarar un principio básico de la IA. Se debe de tener en cuenta que la creación de inteligencia artificial, en este caso por medio de redes

⁵Algunos autores la denotan como función de transferencia.

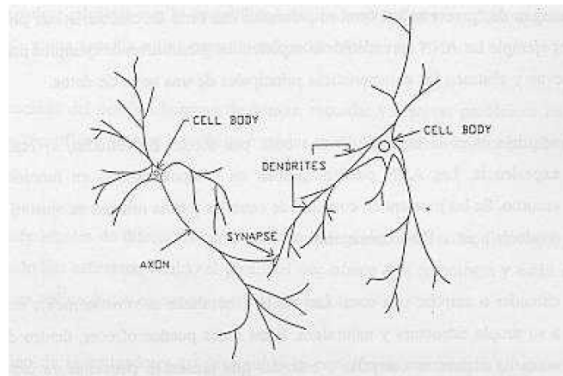


FIGURA 1.2. Componentes de una neurona.
 Fuente: X. B. Olabe, *Redes Neuronales Artificiales y sus Aplicaciones*, (2008).

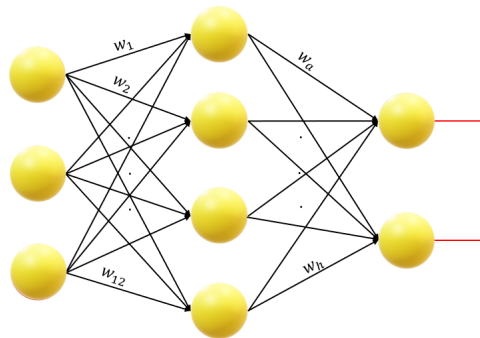


FIGURA 1.3. Componentes de una neurona artificial.
 Fuente: Elaboración propia.

neuronales artificiales, no trata de establecer una competencia con la mente humana, sino de contribuir con ésta a través de la solución de distintas tareas que requieren razonamiento [15]. Tal trabajo se hace con el fin de comprender el comportamiento humano y servir como auxiliar en la realización de funciones complejas.

1.3. Tipos de redes neuronales artificiales

En esta sección se presentan los principales tipos de redes neuronales artificiales que se utilizan en la solución de problemas reales [4]. Se considera la estructura, tipos de funciones de activación y aplicaciones, para cada RNA.

TABLA 1.1. Tipos de estructuras de redes neuronales artificiales.

Nombre	Estructura	Función de activación	Aplicación
Perceptrón (Red prealimentada)	Puede estar integrada por 1 capa o más de 1 capa, convirtiéndose en multicapa. Se introduce información a la red y ésta es asimilada por una determinada regla de aprendizaje para después modificar los parámetros de la red y así poder tener un mejor resultado como salida, respecto al objetivo de la red.	Una función de tipo escalón: $a = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$ donde a denota la evaluación o resultado de la función de activación y n es el argumento de la función de activación.	1. Para clasificar vectores que son linealmente separables o 2. Para resolver problemas de clasificación arbitrarios.
Hamming (Red competitiva)	Utiliza capas prealimentadas y capas recurrentes. El número de neuronas en la primera capa es el mismo que en la segunda capa.	Funciones lineal y lineal positiva, para cada capa respectivamente, definidas como: $a = n$ y $a = \begin{cases} 0, & n < 0 \\ n, & n \geq 0 \end{cases}$ respectivamente.	Para clasificar, para resolver problemas de reconocimiento de patrones binarios.
Hopfield (Red de memoria asociativa recurrente)	Las neuronas son activadas en la capa de entrada y luego la red itera hasta que la capa de salida converja.	Una función lineal saturada simétrica: $a = \begin{cases} -1, & n < -1 \\ n, & n \leq 1 \\ 1, & n > 1 \end{cases}$	Para resolver una gama de problemas de optimización.
Base Radial	Es una red que consiste de 2 capas.	Hay muchos tipos de funciones que se pueden utilizar para la primer capa, por mencionar uno, la función Gaussiana: $a = be^{-\frac{(n-c)^2}{2d^2}}$ para $b, c, d \in \mathbb{R}$ y $d \neq 0$. Para la capa 2, una función lineal estándar (capa lineal estándar).	Para aproximación de funciones o reconocimiento de patrones.

Nombre	Estructura	Función de activación	Aplicación
Grossberg (Red competitiva en tiempo continuo y auto organizada)	Tiene 3 componentes: capa 1, capa 2 y los pesos adaptativos.	Funciones lineal o sigmoide, por ejemplo log-sigmoide: $a = \frac{1}{1+e^{-n}}$	-

Fuente: Elaboración propia con información de [4].

El perceptrón

El perceptrón es una RNA que consiste de una o más entradas (Entrada 1, ..., Entrada n), cada una con sus respectivos pesos (w_1, \dots, w_n), 1 neurona, también llamada procesador, P , y una única salida o . En el procesador se llevan a cabo sumas con las entradas multiplicadas por los pesos (Entrada 1 * w_1 + ... + Entrada n * w_n). Luego, estas sumas se evalúan en una determinada función de activación f , dentro de P , y por último la evaluación de la función $f(W^T E)$ se muestra como la salida o de la red [12]. La función de esta red es llevar a cabo misiones de razonamiento no complejo.

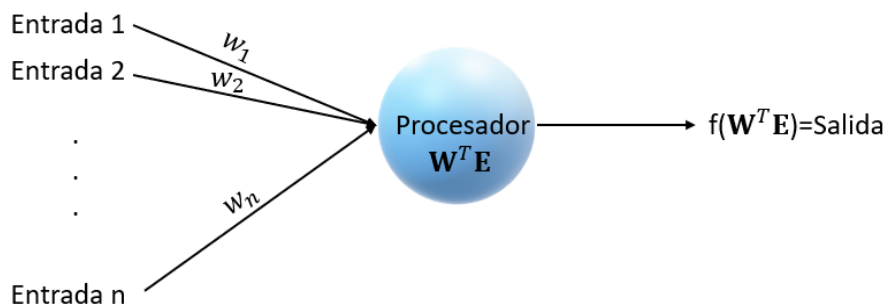


FIGURA 1.4. Un perceptrón con n entradas, n pesos, un procesador o neurona y una salida.

Fuente: Elaboración propia en base a [12].

Este es el modelo neuronal artificial existente más elemental [12].

Dentro de las funciones, es específico, que puede realizarse encuentra la solución a problemas que son linealmente separables y generalización de patrones. Por el lado contrario, dentro de sus limitaciones se encuentra que no puede resolver la función lógica OR-exclusiva y que solo puede aprender funciones lineales [7].

En conclusión, el perceptrón no puede aprender funciones no lineales ni resolver problemas que no son linealmente separables. Ante esta circunstancia, Rosenblatt y Widrow crean las redes de tipo multicapa [4], formando así, la generalización del perceptrón. Con ello se corrigen las limitaciones del perceptrón y por tanto se obtiene

herramienta con un amplio rango de aplicaciones a la vida real. Es así como nace una nueva teoría: el perceptrón multicapa.

El perceptrón multicapa

El perceptrón multicapa es una RNA de tipo *feedforward*, conformada por determinadas neuronas artificiales y capas, que tiene la capacidad de aprender a realizar tareas de tipo intelectual. De manera más formal y específica, se tienen las siguientes definiciones para una neurona y para el perceptrón multicapa:

Una neurona artificial η_i^m , consiste de un operador \sum_i^m , de una función de activación f_i^m , de un bias b_i^m y de una salida o_i^m , donde m = número de la capa de la RNA e i = número de la neurona [4, 15]. Tiene la función de llevar a cabo una pequeña tarea, generada por $o_i^m = f_i^m(\sum_i^m)$, tal que contribuya al objetivo general de la RNA. Observe que en la definición de neurona artificial, el operador \sum_i^m no se definió explícitamente. Esto es porque se requiere de dos elementos más, la matriz de pesos $W_{i,j}^m$ y las entradas z_i , que se encuentran en la siguiente definición sobre el perceptrón multicapa. Así, el operador se define como: $\sum_i^m W_{i,j}^m z_i^m + b_i^m$, donde z_i^m va a representar a las entradas ε_i^m o bien a las salidas de las neuronas o_i^m .

El perceptrón multicapa es una RNA que consiste de al menos una capa oculta, con al menos 2 neuronas η_i^m para cada capa m , donde $i = 1, 2, \dots$, total de neuronas de la capa m y $m = 1, 2, \dots, M$, M denotará la última capa de la red. A la vez, consiste de un conjunto de ε_i entradas, donde cada entrada corresponde a cada neurona η_i^1 . La conexión entre las entradas ε_i y neuronas n_i^1 , en la primera capa, así como la conexión entre las salidas o_i^q , $q \geq 1$, con las neuronas η_i^m , $m \geq 2$, para las capas posteriores, se da por medio de matrices de pesos $W_{i,j}^m$, esto es, una matriz para cada capa de la red. Es un tipo de red de alimentación hacia adelante: *feedforward network* [4, 10]. Tiene la función de llevar a cabo una tarea general, $o^M = f_i^M(\sum_i^M)$, tal que se construye a partir de todas las pequeñas tareas de cada neurona $o^m = f_i^m(\sum_i^m)$, a través de composiciones de funciones sucesivas:

$$o^M = f^M(W^M F^{M-1}(\dots(W^2 f^1(W^1 \varepsilon + b^1) + b^2) + \dots) + b^M)$$

A o^M se le llama la salida de la RNA.

En esta RNA, por ser de tipo *feedforward* la información solo circula en una sola dirección, hacia adelante, o bien hacia la derecha. En otras palabras, no existen ciclos en la red.

El perceptrón multicapa es una RNA llena de conexiones entre cada uno de los elementos que la conforma. Esto es, cada entrada ε_i se conecta con cada neurona de la primera capa η_i^1 , cada salida de la neurona de la primera capa o_i^1 se conecta con cada neurona de la segunda capa η_i^2 y así sucesivamente hasta llegar a las neuronas de la capa de salida η_i^M . Dicho de otra manera, la salida de una capa es la entrada de la siguiente capa, salvo para la primera y última capa, donde sólo se recibe del exterior y sólo se emite al exterior, respectivamente.

Las neuronas pertenecientes a cada capa funcionan en paralelo, esto es, llevan a cabo sus pequeñas tareas al mismo tiempo [4]. Por lo general, las funciones de activación f_i^m de las neuronas las propone el diseñador y no son ajustables i.e. no

cambian, a diferencia de los parámetros escalares de la neurona, como lo son los pesos $w_{i,j}^m$ y los bias b_i^m . Estos son modificados constantemente.

El parámetro bias b_i^m es un parámetro que puede o no influir en la neurona (si su parámetro es distinto a 0 o igual a 0, respectivamente), que se utiliza para que la RNA tenga más probabilidades de convergencia hacia el objetivo. Esto lo hace por medio de la neurona en particular en la que se encuentre [10, 16]. Dicho parámetro siempre tiene un valor de entrada igual a 1 y su parámetro de conexión con la neurona, puede verse como un tipo de peso modificable [17].

Hasta cierto punto, tanto el número de capas ocultas como el número de neuronas en cada una de ellas, dependerá de la naturaleza del problema a solucionar. Esto es porque no existe un método para determinar el número de capas ocultas para la red con sus respectivas neuronas. Lo que se hace al respecto es aplicar prueba y error. En cambio, para determinar el número de entradas y el número de neuronas en la capa de salida, lo que se hace es observar las especificaciones del exterior, es decir, las condiciones del problema que se tenga y de ahí se extraen estos números. La mayoría de las redes neuronales artificiales de tipo multicapa, más realistas o constructivas, a lo más llegan a considerar 2 ó 3 capas ocultas [4]. Sin embargo, también se llega a considerar de forma estándar una sola capa oculta y a lo más 2, para problemas más complicados. Esto se debe a las complicaciones computacionales que se podrían ocasionar [18]. Dependiendo del problema que se tenga, el diseñador trata de estimar dicho número de capas [4].

A continuación se presenta un ejemplo de una RNA.

Ejemplo (Perceptrón multicapa: 3 capas). En la figura 1.5, se observa un perceptrón multicapa con 3 capas. La “capa de entrada” (Inputs), 2 capas ocultas (First Layer y Second Layer) y una capa de salida (Third Layer). En este caso particular se tienen p_i entradas para la RNA, $i = 1, \dots, R$; S^1, S^2 y S^3 neuronas para las 3 capas, respectivamente, 3 matrices de tamaño $S^1 \times R, S^2 \times S^1$ y $S^3 \times S^2$, 3 vectores que representan a los parámetros de bias $\mathbf{b}^1, \mathbf{b}^2$ y \mathbf{b}^3 para cada capa respectivamente, 3 vectores que denotan las funciones de activación $\mathbf{f}^1, \mathbf{f}^2$ y \mathbf{f}^3 , para cada capa, respectivamente, y por último 3 vectores que muestran las salidas de las capas: $\mathbf{a}^1, \mathbf{a}^2$ y \mathbf{a}^3 , respectivamente.

La forma en que la RNA opera es como sigue. Cada entrada p_i , tomada como vector:

$$\mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{pmatrix}$$

se introduce a la red en conjunto con la matriz de pesos \mathbf{W}^1 , la cual está definida explícitamente como:

$$\mathbf{W}^1 = \begin{pmatrix} w_{1,1}^1 & \cdots & w_{1,R}^1 \\ \vdots & \ddots & \vdots \\ w_{S^1,1}^1 & \cdots & w_{S^1,R}^1 \end{pmatrix}$$

luego se ejecuta el operador \sum definido por $\mathbf{W}^1 \mathbf{p} + \mathbf{b}^1$:

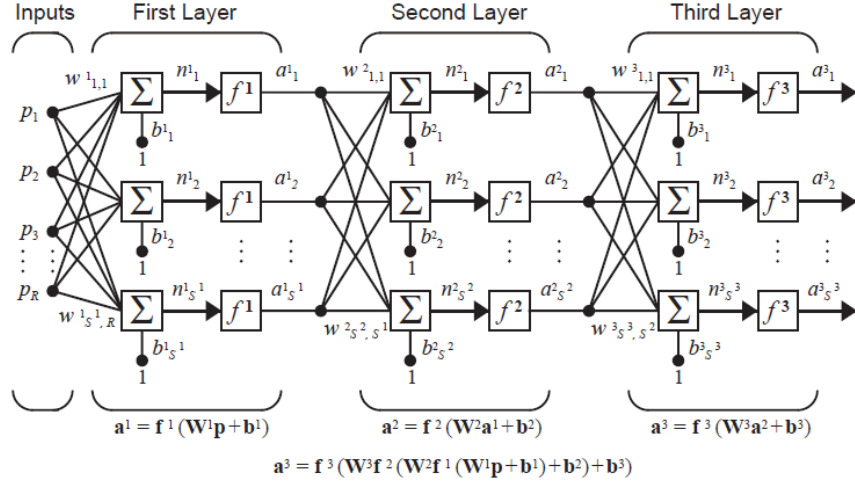


FIGURA 1.5. El perceptrón multicapa con R entradas, 2 capas ocultas con S^1 y S^2 neuronas, respectivamente, y una capa de salida con S^3 neuronas.
Fuente: H. B. Demuth y col., *Neural network design*, Martin Hagan, (2014).

$$\begin{pmatrix} w_{1,1}^1 p_1 + \dots + w_{1,R}^1 p_R + b_1^1 \\ \vdots \\ w_{S^1,1}^1 p_1 + \dots + w_{S^1,R}^1 p_R + b_{S^1}^1 \end{pmatrix}$$

Dicho vector, se le puede llamar vector neto, \mathbf{n}^1 , cuya dimensión es $S^1 \times 1$. Después, \mathbf{n}^1 se traslada al vector \mathbf{f}^1 de funciones de activación para ser evaluado. El resultado de la evaluación se almacena en el vector \mathbf{a}^1 . Este vector \mathbf{a}^1 resulta ser la entrada para la segunda capa. El proceso de entrada y demás ejecuciones en esta segunda capa es análogo al proceso que se llevo a cabo en la primera capa. Lo mismo ocurre para la tercer y última capa. Y como último resultado se obtiene que la salida de la red es el vector \mathbf{a}^3 , de dimensión $S^3 \times 1$, que no es más que la composición de funciones acumulada de las capas anteriores [4]:

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3 \mathbf{a}^2 + \mathbf{b}^3) \quad (1.1)$$

El perceptrón multicapa puede llegar a ser una mejor elección en comparación con las redes de función de base radial, respecto a la memoria de uso y el tiempo de clasificación, por ejemplo, en el problema de identificación de voz. Sin embargo, en cuestiones de tiempo de entrenamiento y en el porcentaje de error (ligeramente más alto), las redes perceptrón multicapa mostraron ser menos efectivas en comparación con las redes de función de base radial [19]. Dentro de otras funciones que puede realizar el perceptrón multicapa se encuentra que distingue datos que no son linealmente separables [20, 21], contrario a lo que hace un perceptrón simple. Por ejemplo, puede obtener o separar datos que pertenecen a una clase A y a una clase B [22]. [4] De forma más general, esta red puede ordenar patrones arbitrarios. También, la RNA tiene la capacidad de aproximar de manera eficiente funciones matemáticas arbitrarias.

Para que el perceptrón multicapa llegue a ejecutar, de manera inteligente, estas

distintas tareas de rango intelectual, se requiere de un algoritmo de entrenamiento para la RNA. En este caso, se emplea un algoritmo de entrenamiento supervisado [10], que se llama *backpropagation* o propagación hacia atrás [4, 23]. El objetivo del algoritmo es que la RNA aprenda a realizar una tarea o a resolver un problema. En términos abstractos, esto se hace por medio de la modificación de los pesos $w_{i,j}^m$ y bias b_i^m de la red, considerando un conjunto específico de entrenamiento (parejas de valores que le indican a la red como debe comportarse). El algoritmo escoge, después de varias iteraciones, los pesos $w_{i,j}^m$ y bias b_i^m ideales, tales que minimizan la diferencia entre los resultados que devuelve la RNA, o_i^M , y los valores que se esperan obtener de la RNA. En otras palabras, la red va aprendiendo de manera autónoma mediante la modificación sucesiva de todos sus pesos $w_{i,j}^m$ y bias b_i^m [4].

En la siguiente sección se analiza el algoritmo *propagación hacia atrás*.

1.4. Algoritmo de entrenamiento para redes neuronales multicapa

Propagación hacia atrás

El perceptrón multicapa es una RNA capaz de clasificar objetos de distintas categorías, también es capaz de encontrar la aproximación de cualquier función matemática a un nivel de exactitud cualesquiera [4] y, como caso particular, puede llevar a cabo pronósticos. Sin embargo, la manera en que la red procede es tipo “prueba y error”. Dado que este procedimiento no es práctico ni tampoco se asegura una solución al problema, entonces se incluye una herramienta extra: un algoritmo de entrenamiento para que la RNA pueda aprender a realizar una tarea de manera eficiente. El algoritmo que se utiliza para entrenar a la red perceptrón multicapa se llama *propagación hacia atrás* [4].

El objetivo que se persigue con este algoritmo es que la RNA aprenda a realizar un trabajo de forma racional. Por ejemplo, aproximar una función⁶ o clasificar patrones [4]. De forma más general, se pretende que la RNA genere salidas que cumplan con las expectativas del diseñador, o que solucione un problema, mediante la actualización de los parámetros de pesos y bias.

La RNA puede arrojar infinitas soluciones para un problema en específico. Dichas soluciones no necesariamente son correctas. Es entonces donde el algoritmo *propagación hacia atrás* interviene. Toma una solución de la red o , la compara con el valor actual que se espera de la red t , y si la diferencia o error $e = t - o$ es distante a 0, entonces vuelve a corregir a la red, de determinada forma, al actualizar sus pesos w y bias b [4]. Estos procedimientos van con el fin de dotar a la RNA de inteligencia artificial.

En la figura 1.6 se presenta un ejemplo que ilustra, a grandes rasgos, cómo interactúa el algoritmo *propagación hacia atrás* en una “RNA”, mediante sus neuronas. En este caso, se supone que el proceso de realización profesional de un dibujo es la tarea general de la RNA. Cada neurona ejecuta una pequeña tarea en específico, así, la primera neurona (capa de entrada) denota la calidad del grafito para un lápiz, la segunda neurona (segunda capa) denota el proveedor, la tercera neurona (tercer capa), la papelería, las tres neuronas consecutivas (cuarta capa) denotan los tipos de lápices,

⁶Se ha demostrado que una red con solo 2 capas, con funciones sigmoide en la capa oculta (con un número considerable de neuronas) y con funciones lineales en la última capa, puede aproximar, virtualmente, cualquier función matemática a cualquier grado de exactitud.

y la última neurona (capa de salida o salida de la red) denota la calificación del dibujo que es de 4.3. Dicha calificación es reprobatoria, la cual se puede interpretar como un error de la RNA. Es aquí donde interviene el algoritmo.

El algoritmo *propagación hacia atrás* empieza a analizar desde la última capa hasta la primera capa para encontrar los errores o sensibilidades que presenta la red y distribuirlos, de forma específica, de atrás hacia adelante. Este proceso se lleva a cabo en conjunto con la actualización de los parámetros de pesos y bias, para cada capa en cuestión, de tal forma que el resultado de la red, vaya tendiendo al valor que se espera de la red: una calificación aprobatoria. En otras palabras, por cada corrección del error o sensibilidad, ocurre una actualización de pesos y bias. Por ejemplo, de 4.3 mejora a 5.5 en la primera iteración del algoritmo, de 5.5 mejora a 5.9 en la segunda iteración, de 5.9 mejora a 7 en la tercera iteración, etc.

Nótese que es un algoritmo recursivo y que una vez que se haya determinado el porcentaje de error para las neuronas de una capa, ésta ahora se toma como si fuese la última capa. El proceso continúa hasta llegar a la primera capa. De aquí, de acuerdo a quien más haya tenido responsabilidad respecto al error final, se asignan los valores de error. En este caso, quien más tuvo responsabilidad fue el tipo de lápiz 2 con 0.9 y quien menos tuvo responsabilidad fue el proveedor con 0.001 [24].

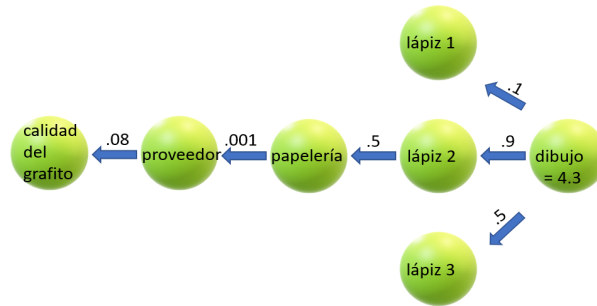


FIGURA 1.6. Ejemplo básico del algoritmo *propagación hacia atrás*. Fuente: Elaboración propia en base a [24].

Después de haber dado un panorama ilustrativo sobre el algoritmo *propagación hacia atrás*, a continuación se introducirá un concepto que define la forma matemática en que se obtienen los parámetros óptimos como son los pesos y bias de la red denominado algoritmo del Descenso del Gradiente Estocástico (DGE), el cual es la esencia del algoritmo propagación hacia atrás.

Teorema (DGE). Considere una RNA multicapa con ε_i entradas, m capas y un coeficiente de aprendizaje α . Sea k el índice que representa a la k -ésima iteración, $W^m(k)$ la matriz de pesos para la capa m , $b^m(k)$ el vector de bias para la capa m , $o(k)$ la salida de la red, $t(k)$ el valor objetivo de la red y $e(k) = t(k) - o(k)$ el error de la RNA. Entonces los parámetros, \mathbf{W} y \mathbf{b} , tales que minimizan el error aproximado de la RNA [4]:

$$\hat{V}(\varepsilon, t, x) = \min_{\mathbf{W}, \mathbf{b}} (\mathbf{t}(k) - \mathbf{o}(k))^T (\mathbf{t}(k) - \mathbf{o}(k)) = \min_{\mathbf{W}, \mathbf{b}} \mathbf{e}(k)^T \mathbf{e}(k),$$

en la iteración k , son:

$$\mathbf{W}^m(k+1) - \mathbf{W}^m(k) = -\alpha \frac{\partial \hat{V}}{\partial \mathbf{W}^m} \quad (1.2)$$

$$\mathbf{b}^m(k+1) - \mathbf{b}^m(k) = -\alpha \frac{\partial \hat{V}}{\partial \mathbf{b}^m} \quad (1.3)$$

donde el parámetro x , en \hat{V} , representa todos los pesos y bias de la red, y las parciales $\frac{\partial \hat{V}}{\partial \mathbf{W}^m}$ y $\frac{\partial \hat{V}}{\partial \mathbf{b}^m}$, denotan la variación de la aproximación del error cuando varían los parámetros de pesos \mathbf{W}^m y bias \mathbf{b}^m , respectivamente [4, 10]. Dicho esto, ya se puede proceder a establecer la definición del algoritmo *propagación hacia atrás*.

Considere una RNA multicapa con m capas, donde $m = 1, 2, \dots, M-1, M$. Sea $\{(\varepsilon_1, t_1), (\varepsilon_2, t_2), \dots, (\varepsilon_n, t_n)\}$ un conjunto finito de entrenamiento para la red, donde ε_i denota los inputs o entradas de la red y t_i denota los valores que se esperan obtener de la red, \mathbf{W}^m la matriz de pesos para cada capa m , \mathbf{b}^m el vector que denota los bias, \mathbf{o}^m el vector que contiene los valores de salida de la red, \mathbf{n}^m el neto u operador \sum , \mathbf{F}^m la matriz diagonal cuyos elementos son las funciones de activación, \mathbf{S}^m el número de neuronas en la capa m y $\mathbf{e} = \mathbf{t} - \mathbf{o}$ el error de la red. Además, considere las hipótesis del teorema (DGE). Luego el algoritmo *propagación hacia atrás* minimiza el Error Cuadrático Medio (ECM) de la RNA [4]:

$$V(\varepsilon, t, x) = \min_{\mathbf{W}, \mathbf{b}} E[(\mathbf{t} - \mathbf{o})^T (\mathbf{t} - \mathbf{o})] = \min_{\mathbf{W}, \mathbf{b}} E[\mathbf{e}^T \mathbf{e}] \quad (1.4)$$

a través de k aproximaciones:

$$\hat{V}(\varepsilon, t, x) = \min_{\mathbf{W}, \mathbf{b}} (\mathbf{t}(k) - \mathbf{o}(k))^T (\mathbf{t}(k) - \mathbf{o}(k)) = \min_{\mathbf{W}, \mathbf{b}} \mathbf{e}(k)^T \mathbf{e}(k) \quad (1.5)$$

que a su vez utiliza el algoritmo del descenso del gradiente estocástico, teorema (DGE), para encontrar los valores óptimos \mathbf{W}^m y \mathbf{b}^m . Dichos valores son [4, 10]:

$$\mathbf{W}^m(k+1) - \mathbf{W}^m(k) = -\alpha \mathbf{s}^m (\mathbf{o}^{m-1})^T \quad (1.6)$$

$$\mathbf{b}^m(k+1) - \mathbf{b}^m(k) = -\alpha \mathbf{s}^m \quad (1.7)$$

donde

$$\mathbf{s}^m = \begin{cases} \frac{\partial \mathbf{F}^m(\mathbf{n}^m)}{\partial \mathbf{n}^m} (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, & m = M-1, \dots, 2, 1 \\ -2 \frac{\partial \mathbf{F}^m(\mathbf{n}^m)}{\partial \mathbf{n}^m} (\mathbf{t} - \mathbf{o}), & m = M \end{cases}$$

Con la observación de que el punto de inicio de \mathbf{s}^m , la sensibilidad, es en M , y a partir de ahí se calcula el resto de sensibilidades de forma recursiva. La sensibilidad \mathbf{s} mide el cambio que se produce en la aproximación del ECM, \hat{V} , respecto a la variación del resultado del operador \sum que es el neto $\mathbf{n} = \mathbf{W}\mathbf{z} + \mathbf{b}$, donde \mathbf{z} representa las entradas ε o las salidas de las capas \mathbf{o}^m [4].

Observe que en la definición *propagación hacia atrás* se han omitido los cálculos,

por cuestiones de claridad, para la obtención de los parámetros óptimos del algoritmo, a partir del (DGE). Dichos cálculos se realizan a continuación [4].

Se verá que

$$\frac{\partial \hat{V}}{\partial \mathbf{W}^m} = \mathbf{s}^m (\mathbf{o}^{m-1})^T \quad (1.8)$$

y

$$\frac{\partial \hat{V}}{\partial \mathbf{b}^m} = \mathbf{s}^m \quad (1.9)$$

$$\text{donde } \mathbf{s}^m = \begin{cases} \frac{\partial \mathbf{F}^m(\mathbf{n}^m)}{\partial \mathbf{n}^m} (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, & m = M-1, \dots, 2, 1 \\ -2 \frac{\partial \mathbf{F}^m(\mathbf{n}^m)}{\partial \mathbf{n}^m} (\mathbf{t} - \mathbf{o}), & m = M \end{cases}$$

Primero observe lo siguiente. Las componentes i del vector neto para la capa m , n_i^m , están definidas explícitamente como [4]:

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m o_j^{m-1} + b_i^m \quad (1.10)$$

Luego, si se calcula la variación de la componente i del vector neto, correspondiente a la capa $m+1$, respecto a la variación de la componente j del vector neto, correspondiente a la capa m , se tiene [4]:

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = \frac{\partial \left(\sum_{j=1}^{S^m} w_{i,j}^{m+1} o_j^m + b_i^{m+1} \right)}{\partial n_j^m} = \frac{\partial \left(\sum_{j=1}^{S^m} w_{i,j}^{m+1} f^m(n_j^m) + b_i^{m+1} \right)}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} \quad (1.11)$$

Ahora se calculan las parciales $\frac{\partial \hat{V}}{\partial w_{i,j}^m}$ y $\frac{\partial \hat{V}}{\partial b_i^m}$ pertenecientes al algoritmo (DGE) [4]:

$$\frac{\partial \hat{V}}{\partial w_{i,j}^m} = \frac{\partial \hat{V}}{\partial n_i^m} \cdot \frac{\partial n_i^m}{\partial w_{i,j}^m} \quad (1.12)$$

$$= \frac{\partial \hat{V}}{\partial n_i^m} \cdot o_j^{m-1} \quad (1.13)$$

$$= \left(\frac{\partial n_i^{m+1}}{\partial n_j^m} \right)^T \frac{\partial \hat{V}}{\partial n_i^{m+1}} \cdot o_j^{m-1} \quad (1.14)$$

$$= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} \frac{\partial \hat{V}}{\partial n_i^{m+1}} \cdot o_j^{m-1} \quad (1.15)$$

$$\implies \frac{\partial \hat{V}}{\partial w_{i,j}^m} = \frac{\partial \hat{V}}{\partial n_i^m} \cdot o_j^{m-1} = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} \frac{\partial \hat{V}}{\partial n_i^{m+1}} \cdot o_j^{m-1} \quad (1.16)$$

y

$$\frac{\partial \hat{V}}{\partial b_i^m} = \frac{\partial \hat{V}}{\partial n_i^m} \cdot \frac{\partial n_i^m}{\partial b_i^m} \quad (1.17)$$

$$= \frac{\partial \hat{V}}{\partial n_i^m} \cdot 1 \quad (1.18)$$

$$= \left(\frac{\partial n_i^{m+1}}{\partial n_j^m} \right)^T \frac{\partial \hat{V}}{\partial n_i^{m+1}} \quad (1.19)$$

$$= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} \frac{\partial \hat{V}}{\partial n_i^{m+1}} \quad (1.20)$$

$$\implies \frac{\partial \hat{V}}{\partial b_i^m} = \frac{\partial \hat{V}}{\partial n_i^m} \cdot 1 = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} \frac{\partial \hat{V}}{\partial n_i^{m+1}} \quad (1.21)$$

Si se denota

$$s_i^{m+1} = \frac{\partial \hat{V}}{\partial n_i^{m+1}} \quad (1.22)$$

las ecuaciones (1.16) y (1.21) se denotan, respectivamente, como [4]:

$$\frac{\partial \hat{V}}{\partial w_{i,j}^m} = s_i^m \cdot o_j^{m-1} = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} s_i^{m+1} \cdot o_j^{m-1} \quad (1.23)$$

y

$$\frac{\partial \hat{V}}{\partial b_i^m} = s_i^m \cdot 1 = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} s_i^{m+1} \quad (1.24)$$

De ambas ecuaciones se deduce que [4]:

$$s_i^m = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} s_i^{m+1} \quad (1.25)$$

Así, al sustituir las ecuaciones (1.23) y (1.24) en el algoritmo del descenso del gradiente estocástico, ecuaciones (1.2) y (1.3), respectivamente, se tiene [4]:

$$w_{i,j}^m(k+1) - w_{i,j}^m(k) = -\alpha s_i^m \cdot o_j^{m-1} \quad (1.26)$$

$$b_i^m(k+1) - b_i^m(k) = -\alpha s_i^m \quad (1.27)$$

donde $s_i^m = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} s_i^{m+1}$.

Dicho esto, ya se sabe cómo se derivan los s_i^m , solo resta encontrar el último elemento s_i^M , de la última capa, para poder encontrar cada s_i^m , $m = 1, \dots, M - 1$ restante. Esto es, cada sensibilidad es propagada hacia atrás, de ahí el nombre de *propagación hacia atrás* [4]:

$$s_i^M = \frac{\partial \hat{V}}{\partial n_i^M} \quad (1.28)$$

$$= \frac{\partial (\mathbf{t} - \mathbf{o})^T (\mathbf{t} - \mathbf{o})}{\partial n_i^M} \quad (1.29)$$

$$= \frac{\partial \sum_{j=1}^{S^M} (t_j - o_j)^2}{\partial n_i^M} \quad (1.30)$$

$$= \frac{\partial \sum_{j=1}^{S^M} (t_j - f_j(n))^2}{\partial n_i^M} \quad (1.31)$$

$$= -2(t_i - f_i(n)) \frac{\partial f_i(n)}{\partial n_i^M} \quad (1.32)$$

O bien, $s_i^M = -2(t_i - o_i) \frac{\partial f_i(n)}{\partial n_i^M}$.

Reescribiendo los resultados (1.26) y (1.27), junto con las ecuaciones de sensibilidad s^m y s^M , en notación matricial queda [4]:

$$\mathbf{W}^m(k+1) - \mathbf{W}^m(k) = -\alpha \mathbf{s}^m (\mathbf{o}^{m-1})^T \quad (1.33)$$

$$\mathbf{b}^m(k+1) - \mathbf{b}^m(k) = -\alpha \mathbf{s}^m \quad (1.34)$$

donde

$$\mathbf{s}^m = \begin{cases} \frac{\partial \mathbf{F}^m(\mathbf{n}^m)}{\partial \mathbf{n}^m} (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, & m = M - 1, \dots, 2, 1 \\ -2 \frac{\partial \mathbf{F}^m(\mathbf{n}^m)}{\partial \mathbf{n}^m} (\mathbf{t} - \mathbf{o}), & m = M \end{cases}$$

Por lo tanto se tiene (1.8) y (1.9), es decir, los parámetros óptimos (1.6) y (1.7) del algoritmo *propagación hacia atrás* [4].

En última instancia el algoritmo busca que $\mathbf{s}^m \rightarrow 0$. Esto se debe a que, al no haber sensibilidad \mathbf{s}^m , no hay cambio entre la función \hat{V} y el neto \mathbf{n}^m , las derivadas parciales serían 0, con ello se alcanzan los pesos \mathbf{W}^m y bias \mathbf{b}^m óptimos tales que minimizan el ECM aproximado \hat{V} , es decir, con ello se tendría $\mathbf{o} \approx \mathbf{t}$ [4].

En la práctica, el algoritmo requiere de un conjunto finito de entrenamiento. Este le sirve a la red como ejemplo de comportamiento a seguir [4]:

$$\{(\varepsilon_1, t_1), (\varepsilon_2, t_2), \dots, (\varepsilon_n, t_n)\}$$

Cada ε_i se corresponde con cada t_i . Por ejemplo, suponga que solo se considera a una pareja, tome (ε_1, t_1) , entonces la red recibirá como entrada a ε_1 , hará determinados cálculos, en base a éste, y al final la RNA devolverá una salida o_1 tal que se comparará con t_1 [4].

Luego, para comenzar con la ejecución de la RNA, y posteriormente con la del algoritmo, se requiere de un conjunto inicial arbitrario de pesos, usualmente los valores se consideran en $(-0.5, 0.5)$, o $(-1, 1)$ o $\left(-\frac{1}{\text{num. pesos}}, \frac{1}{\text{num. pesos}}\right)$ [18] tales que conformarán las matrices \mathbf{W}^m para cada capa. De forma similar también se requiere de un conjunto inicial de bias arbitrarios \mathbf{b}^m , para cada capa, y de un coeficiente de aprendizaje α , que generalmente se considera en el intervalo $(0.25, 0.75)$ [10]; o también dentro del intervalo $(.01, 0.99)$, y si se considera éste último, usualmente se toma $\alpha = 0.1$ [18]. Con esto presente, se realiza la primer ejecución del perceptrón multicapa [4].

Una vez que la red haya devuelto su resultado o_i^M , empieza la primer iteración del algoritmo *propagación hacia atrás* [4].

El o_i^M , se compara, uno a uno, con cada valor objetivo que se espera de la red t_i . Si esta comparación es un valor lejano a 0, digamos 1.3, se empieza a hacer un análisis desde la última capa hasta la primer capa, mediante la sensibilidad de la función del ECM aproximado \hat{V} , respecto al valor neto n_i^m , es decir, un análisis a través de las derivadas parciales $\frac{\partial \hat{V}}{\partial n_i^m}$. Se calcula el valor de la parcial $s_i^m = \frac{\partial \hat{V}}{\partial n_i^m}$, con la fórmula explícita $\mathbf{s}^M = -2 \frac{\partial \mathbf{F}^M(\mathbf{n}^M)}{\partial \mathbf{n}^M}(\mathbf{t} - \mathbf{o})$, para la última capa y $\mathbf{s}^m = \frac{\partial \mathbf{F}^m(\mathbf{n}^m)}{\partial \mathbf{n}^m} (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}$, para $m = M - 1, \dots, 2, 1$, el resto de las capas previas [4]. A la vez, se espera que estas sensibilidades vayan presentando una tendencia de acercamiento a 0. Esta convergencia, mide el aprendizaje de la RNA [10].

Una vez calculadas todas las sensibilidades para cada una de las capas de la red, se sustituyen en las ecuaciones (1.6) y (1.7), para poder actualizar todas las matrices de pesos \mathbf{W}^m y vectores bias \mathbf{b}^m de cada capa. Esto es con el fin de poder mejorar la intervención de la RNA [4].

Hasta aquí, todo lo que se describió fue para la primer iteración, una única iteración. El proceso se repite k veces, generalmente utilizando todos los elementos del conjunto de entrenamiento y repitiéndolos, hasta que se obtenga casi la igualdad entre o_i^M y t_i [4]. Sin duda es un proceso que tomaría bastante tiempo si se realizará a mano y no sería eficiente. Es por ello que se hace uso de softwares como Matlab para elaborar códigos tales que lleven a cabo extensas iteraciones que requieren de álgebra lineal y cálculo diferencial en varias variables. En el capítulo 3, se estudiará e implementará la programación de la red perceptrón multicapa con el *propagación hacia atrás* como su algoritmo de entrenamiento.

1.5. Aplicaciones

Basogain menciona en [10] que “algunas de las áreas de aplicación de las redes neuronales artificiales son: Análisis y Procesado de señales, Reconocimiento de Imágenes, Control de Procesos, Filtrado de ruido, Robótica, Procesado del Lenguaje, Diagnósticos médicos, Modelado y Pronóstico, Otros”.

Dentro del área financiera, se expresan las siguientes aplicaciones de redes neuronales: “en la administración de los portafolios de inversión, la evaluación de acciones y bonos, las estrategias de cobertura y arbitraje, el análisis de la volatilidad y correlación y la predicción de los precios de acciones, bonos, tasas de interés y de cambio, entre otros” [14]. “La mayoría de los temas relacionados tienen que ver con la predicción de quiebra o la predicción de rendimientos de valores, que ayuden a tomar mejores decisiones financieras. Asimismo, las Redes Neuronales pueden emplearse como herramientas complementarias a los enfoques tradicionales de análisis multivariante.” [14]

Se ha dado una introducción sobre las redes neuronales artificiales, en particular se trató a la red perceptrón multicapa. Se analizó la forma en que la red puede lograr aprender, mediante el algoritmo propagación hacia atrás.

En el siguiente capítulo, se analizan conceptos básicos sobre la inflación, con el INPC.

CAPÍTULO 2

La economía, la inflación y su pronóstico

En este capítulo se analiza uno de los fenómenos económicos más relevantes, tanto de una economía cerrada como global. Se trata del fenómeno denominado inflación. La inflación juega un papel clave para determinar las decisiones de consumo que influyen en la calidad de vida de los distintos agentes económicos: familias, empresas y gobierno.

Del conocimiento, y del comportamiento de la inflación dependen decisiones: ¿Cómo modificar o crear una nueva política monetaria?, ¿Cuánto y cuando conviene más invertir en una empresa? o ¿Por qué se debería de limitar el gasto personal?. Decisiones que, dependiendo de su elección, dejan determinada una buena parte de la vida de todos los ciudadanos.

En el presente capítulo se estudia el fenómeno de la inflación mediante el INPC. Se revisa la construcción, validación y aplicación del INPC. El capítulo se concluye con el análisis sobre métodos de pronóstico, aplicados a distintos tipos de datos, uno de ellos el INPC.

2.1. La inflación

La inflación se define como un incremento, "generalizado y sostenido"[25], de los distintos precios de bienes y servicios, de un determinado país [25]; y a "la variación porcentual del nivel general de precios" se le llama la tasa de inflación [26] y se denotará como $\pi(t)$, donde t representa el tiempo para el cual se obtiene la tasa de inflación. Para calcular dicha tasa de inflación, en términos del INPC, primero se elige un año base, tome t , junto con su $INPC_t$ respectivo, luego se escoge el año con el cual se va a comparar, digamos t' , con su respectivo $INPC_{t'}$, así [27, 28]:

$$\pi(t') = \left(\frac{INPC_{t'} - INPC_t}{INPC_t} \right) \cdot 100 = \% \text{ de la tasa de inflación respecto al año } t \quad (2.1)$$

Otra forma de encontrar la tasa de inflación es mediante el deflactor del Producto Interno Bruto (PIB). El deflactor del PIB es un índice que mide la relación que hay entre el PIB nominal y PIB real. Este muestra por cuanto se incrementa (disminuye) el precio de la producción, en un determinado periodo de tiempo t .

El deflactor del PIB en el año t se define como:

$$P_t = \frac{PIB_t \text{ nominal}}{PIB_t \text{ real}} \cdot 100 \quad (2.2)$$

De aquí que indica el precio promedio de la producción en la economía, en el sentido de que se toman en cuenta todos los precios de los productos de una canasta representativa.

Observe que la tasa de variación de P_t es $\pi_d = \left(\frac{P_t - P_{t-1}}{P_{t-1}} \right) \cdot 100$ e indica la tasa porcentual a la que incrementa el nivel general de precios respecto al año $t - 1$, es decir, representa la tasa de inflación.

Así, se tienen dos formas de calcular la tasa de inflación, mediante el INPC o mediante el deflactor del PIB P_t [29].

La diferencia entre la inflación y el deflactor del PIB es que la inflación involucra solo la variabilidad del índice de precios al consumidor, de un país, durante un determinado periodo de tiempo; y el deflactor del PIB, considera el cambio en los precios, pero de la producción en su total¹, es decir, no solamente elige los precios de la producción adquirida por parte de los consumidores, sino todos los precios dentro de una economía.

Para la inflación, quién tiene el cargo para dirigirla es el Banco de México. El Banco de México tiene la función de mantener el poder adquisitivo de la moneda con ayuda de los mecanismos de la política monetaria. Desde el año 2003 la política monetaria tiene el objetivo de controlar la inflación en un dígito. La política inicio con una meta anual del INPC de 3%, con un intervalo de variación de $\pm 1\%$. En la actualidad, el rango considerado por el Banco de México para la inflación es de 2% a 4% [30]. Y quien tiene la responsabilidad de dar a conocer la inflación o de emitir los informes sobre la inflación, periódicamente, es el INEGI. Esto sucedió a partir de la primera quincena de julio de 2011, en la dirección [31].

Inflación subyacente

La inflación subyacente es un caso particular de la inflación, en la cual el INPC involucrado no considera determinados bienes y servicios de la canasta representativa. Los precios de los bienes y servicios que sí se consideran en este INPC, son precios independientes de la estacionalidad (e.g. alteraciones por plagas), la alta volatilidad (e.g. productos que están en el mercado mundial, y que su precio varía debido al cambio de moneda), y de las decisiones administrativas por parte del gobierno (e.g. aumento de la gasolina) [32].

La elección de este tipo específico de inflación fue precisamente por su definición, dado que de esta forma se esperaría tener una mejor aproximación a los distintos cambios en los precios del INPC, de forma uniforme, así, se tendría una menor sensibilidad al error o a equivocarse, al momento de predecir [32].

¹Que a su vez involucra la parte nominal y real del PIB.

Inflación no subyacente

La inflación no subyacente es la inflación que involucra al INPC que considera los bienes y servicios de una canasta representativa de los consumidores tales que no pertenecen a la canasta del INPC subyacente. Es el complemento de la subyacente.

En la siguiente sección se analiza de manera detallada el INPC.

2.2. El Índice Nacional de Precios al Consumidor (INPC)

En México, el INPC es un indicador económico que mide la proporción del cambio de precios de una canasta representativa de consumo por las familias, a través del tiempo [25]. Este también se puede definir como una medida promedio del nivel de precios de bienes y servicios adquiridos por los consumidores en un país, durante un período de tiempo, es decir, se puede interpretar como una medida del coste de la vida de los consumidores [29, 28]. Este índice, es la base para conocer cómo se comporta y mide la inflación en una economía [25].

Antecedentes del INPC

Antes de establecerse el INPC como indicador de la inflación en México, en 1823 el periodista y economista político, Joseph Lowe (x-1831) [33] propuso la primera definición de los índices de precios², el cual recibió el nombre de índice de Lowe, por respetabilidad [34]. En esta tesis el índice de Lowe se denotará como I_{Lw} .

En general, las instituciones donde llevan a cabo estadísticas de economía o material con fines didácticos, utilizan alguna variante del índice de Lowe [34, 28]. Esto se verá más adelante cuando se defina el índice uniforme y el índice de Laspeyres. A continuación se expone la definición I_{Lw} .

Sea X un conjunto de bienes y servicios, arbitrario, adquiridos por los consumidores de un país. Sea $x \in X$ un producto, p_x^s el precio del producto x en el tiempo s , p_x^t el precio del producto x en el tiempo t y q_x la cantidad del producto x adquirida en cualquier tiempo, donde $s, t, p_x^s, p_x^t, q_x \in \mathbb{R}$ y $s < t$. Entonces el índice de Lowe, I_{Lw} , se define como [34]:

$$I_{Lw} = \sum_{x \in X} \frac{p_x^t q_x}{p_x^s q_x} \quad (2.3)$$

En la ec. (2.3) de antemano se asume que $\sum_{x \in X} p_x^s q_x \neq 0$. El índice indica o mide la variación proporcional de los precios y/o cantidades, del tiempo t respecto al tiempo s , sobre los productos de una canasta arbitraria X . Es una clase de índices de precios muy conocida debido a que es intuitiva [34] en relación a lo que quiere representar, esto es porque el cociente o índice I_{Lw} refleja por cuanto cambia el total de productos adquiridos, considerando únicamente el tiempo específico t de los precios o $\sum_{x \in X} p_x^t q_x$

²no confundir con la creación de los primeros índices de precios, los cuales surgen en el siglo XVIII [34].

respecto al total de productos adquiridos, considerando únicamente el tiempo específico s de los precios o $\sum_{x \in X} p_x^s q_x$.

I_{LW} es una clase de índices abstracta. Observe que la ec. (2.3) puede involucrar 3 tiempos distintos [34]:

1. El tiempo en el cual se lleva a cabo la encuesta para construir el conjunto X .
2. El tiempo en el cual se define el conjunto X .
3. El tiempo en el cual se calcula el índice I_{LW} (mensual, quincenal, etc.).

Por lo tanto, la definición de Lowe brinda una amplia gama de posibilidades para el cálculo del índice. No obstante, el índice no necesariamente provee la información que se espera de un índice, pues no se sabe si los resultados del índice son debido a cambios en los precios o a cambios en las cantidades [34]. Una variación de I_{LW} , tal que acota su alcance, en el siguiente sentido: la cantidades que se compran en el tiempo k , se pagan al tiempo k , es la siguiente:

Sea X un conjunto de bienes y servicios, arbitrario, adquirido por los consumidores de un país. Sea $x \in X$ un producto, p_x^s el precio del producto x en el tiempo s , p_x^t el precio del producto x en el tiempo t , q_x^s la cantidad del producto x adquirida en el tiempo s y q_x^t la cantidad del producto x adquirida en el tiempo t , donde $s, t, p_x^s, p_x^t, q_x^s, q_x^t \in \mathbb{R}$ y $s < t$. Entonces el índice uniforme, I_u , se define como [35, 36, 28]:

$$I_u = \sum_{x \in X} \frac{p_x^t q_x^t}{p_x^s q_x^s} \quad (2.4)$$

En la ec. (2.4) la expresión $\sum_{x \in X} p_x^t q_x^t$ denota el valor total de las transacciones en el mercado, al tiempo t , y la expresión $\sum_{x \in X} p_x^s q_x^s$ denota el valor total de las transacciones en el mercado, al tiempo s .

Observe que en (2.4) se encuentra que representa, hasta cierto punto, una buena³ aproximación para calcular el índice dado que las cantidades adquiridas en el tiempo k se pagan al tiempo k , tanto en el numerador como en el denominador y a su vez, el cociente expresa la variación de los precios y/o cantidades en el tiempo t respecto al tiempo s . Sin embargo, I_u , así como I_{LW} , tiene la deficiencia de que si llega a aumentar o a disminuir el índice, no se sabrá si fue debido al cambio en los precios o al cambio en las cantidades respecto a ambos tiempos s y t . Así, se introduce una nueva variante, con el intento de solucionar dicho problema. Se trata del índice de Laspeyres, I_{Lpy} , en honor al economista Ernst Louis Étienne Laspeyres (1834-1913) [37, 34]. Se enuncia como sigue [34]:

Sea X un conjunto de bienes y servicios, arbitrario, adquirido por los consumidores de un país. Sea $x \in X$ un producto, p_x^s el precio del producto x en el tiempo s , p_x^t el precio del producto x en el tiempo t y q_x^s la cantidad del producto x adquirida en el tiempo s , donde $s, t, p_x^s, p_x^t, q_x^s \in \mathbb{R}$ y $s < t$. Entonces el índice de Laspeyres, I_{Lpy} , se define como:

³Depende de las necesidades, fines y/o punto de vista de cada investigador [34].

$$I_{Lpy} = \sum_{x \in X} \frac{p_x^t q_x^s}{p_x^s q_x^t} \quad (2.5)$$

La ec. (2.5) denota la variación proporcional de las transacciones considerando la canasta X , en el tiempo t respecto al tiempo s , tomando el periodo de referencia de los precios igual al periodo de referencia de las cantidades. Note que el I_{Lp} a diferencia de I_{Lw} o I_u , sí devuelve lo que se espera de un índice de precios: la variación proporcional de los precios y no de las cantidades.

Para que un índice sea de tipo Laspeyres, éste tiene que considerar el mismo tiempo para sus elementos que lo conforman. Es decir, el tiempo de referencia de las ponderaciones, el tiempo de referencia de los precios y el tiempo de referencia del índice, debe ser el mismo tiempo [38].

El índice de Laspeyres es preferido por distintas agencias estadísticas respecto a otros tipos de índices (e.g. el índice de Paasche) debido a que requiere de menos tiempo y esfuerzo para calcularlo. Esto es porque no es necesario conocer las cantidades del producto x al tiempo actual t , sino que es suficiente con conocer las cantidades del producto x al tiempo de referencia s [39]. Además, el índice I_{Lpy} es una medida de cálculo de estándar internacional, que de manera general es utilizada en todo el mundo, así, también la utilizan países desarrollados. Tal índice I_{Lpy} permite comparar los índices de precios en diferentes países [38]. Y por último, el índice I_{Lpy} es el escogido en México para representar el índice de precios de los consumidores.

Validación de índice de precios

Una forma básica⁴ de ver de manera clara el comportamiento, ventajas y desventajas, de los distintos tipos de índices de precios, es mediante los nueve tests o pruebas, presentes en esta sección [41]. Por ejemplo, el índice de Jevons, que se utiliza indirectamente para el cálculo del INPC, satisface todas las pruebas. Sin embargo, el índice de Laspeyres, que se utiliza directamente para el cálculo del INPC, satisface solo 7 de éstas (1,2,3,4,6,7 y 8) [40].

Sea $p^i \in \mathbb{R}^n$ un vector que representa los precios en el periodo i , $q^i \in \mathbb{R}^n$ un vector que representa las cantidades en el periodo i , para $i = 0, 1, 2$. Sea $I(p^0, p^1, q^0, q^1)$ un índice de precios arbitrario que representa el nivel de precios en el periodo 1 relativo al periodo base 0. Entonces el índice de precios I satisface las siguientes pruebas [40]:

1. Prueba de la identidad

$$\text{Si } p^0 = p^1 \text{ y } q^0 = q^1, I(p^0, p^1, aq^0, bq^1) = 1, \forall a, b \in \mathbb{R}^+$$

2. Prueba de la proporcionalidad

$$I(p^0, ap^1, q^0, q^1) = aI(p^0, p^1, q^0, q^1), \forall a \in \mathbb{R}^+$$

3. Prueba de la invarianza ante cambios escala

$$I(ap^0, ap^1, bq^0, cq^1) = I(p^0, p^1, q^0, q^1), \forall a, b, c \in \mathbb{R}^+$$

⁴Existen más tests. Así, es posible que el estatus actual de los índices cambie, ya sea que mejore o empeore. Además hay otras técnicas (por ejemplo desde un acercamiento económico e.g. microeconómico) para evaluar a un índice de precios [40].

4. Prueba de conmensurabilidad

$$I(ap^0, ap^1, a^{-1}q^0, a^{-1}q^1) = I(p^0, p^1, q^0, q^1), \forall a \in \mathbb{R}^n \text{ y } a > 0$$

5. Prueba del tratamiento simétrico de países o tiempos

$$I(p^1, p^0, q^1, q^0) = \frac{1}{I(p^0, p^1, q^0, q^1)}$$

6. Prueba del tratamiento simétrico de productos (mercancías)

$$I(\hat{p}^0, \hat{p}^1, \hat{q}^0, \hat{q}^1) = I(p^0, p^1, q^0, q^1)$$

donde \hat{p}^i y \hat{q}^i denotan las mismas permutaciones de los elementos de p^i y q^i , respectivamente.

7. Prueba de monotocidad

$$p^1 \leq p^2 \implies I(p^0, p^1, q^0, q^1) \leq I(p^0, p^2, q^0, q^2)$$

8. Prueba del valor medio

$$\min_n \frac{p^1}{p^0} \leq I(p^0, p^1, q^0, q^1) \leq \max_n \frac{p^1}{p^0}$$

9. Prueba de circularidad

$$I(p^0, p^1, q^0, q^1)I(p^1, p^2, q^1, q^2) = I(p^0, p^2, q^0, q^2)$$

Las pruebas se interpretan como sigue [40]:

1. Si los precios y las cantidades de los bienes y servicios en consideración no cambian de un periodo a otro (0 a 1), entonces el índice, considerando las cantidades para ambos periodos multiplicadas por cualquier constante positiva (no necesariamente iguales), es igual a 1.
2. Un índice que considera a su precio actual multiplicado por una constante positiva arbitraria a , es igual al índice multiplicado por a .
3. Si los precios de los periodos 0 y 1 se multiplica por un mismo número positivo y las cantidades de los periodos 0 y 1 se multiplican por números positivos arbitrarios, entonces el índice permanece constante. Esto es, el índice alterado por estas constantes a , b y c , es igual al índice original.
4. Si los precios de ambos periodos se multiplican por una misma constante positiva a y las cantidades de ambos periodos se multiplican por el inverso multiplicativo de a , entonces el índice se mantiene constante, respecto al índice original.
5. Si se intercambian los papeles de los periodos 0 y 1 en el índice, entonces éste índice debe ser igual al recíproco del índice original. En otras palabras, si se invierten los tiempos, y ahora se va en retroceso, de 1 a 0, entonces se tiene un índice de precios tipo I^{-1} .
6. Indica que el índice de precios puede considerar sus argumentos en cualquier orden, estos siguen siendo los mismos, y por tanto el resultado final no se altera.
7. Si el precio de un periodo 1 es menor o igual al precio del periodo 2, entonces el índice que compara los precios respecto a los periodos 0 y 1 es menor o igual al índice que compara los precios respecto a los periodos 0 y 2.
En otras palabras, si $p^2 \geq p^1$, entonces los índices que involucran estos precios, preservan esta condición.

8. Cualquier índice de precios debe de pertenecer al intervalo que considera el mínimo y máximo de los radios de precios de todos los productos n : $\left[\min_n \frac{p^1}{p^0}, \max_n \frac{p^1}{p^0} \right]$
9. Indica la transitividad existente en los índices, al considerar 3 periodos distintos: 0, 1, 2. Estos es, el producto del índice del periodo 0 al 1, con el índice del periodo 1 al 2, es igual al índice del periodo que va del 0 al 2.

2.3. Metodología de cálculo y construcción del INPC

En esta sección se presenta la manera en que se construye el INPC, en México. Se menciona desde los elementos que el índice requiere y la forma en que estos se obtienen, hasta la descripción de su cálculo. Luego se justifica la validez del índice. La sección se finaliza con ejemplos de aplicación del INPC a la vida de los agentes económicos, mediante la toma de decisiones; y a su vez, se menciona quién lleva el control (recopilación de datos, procesado de datos y divulgación del índice) sobre el INPC y el porqué de dicha asignación, a la respectiva autoridad.

Para el cálculo del INPC se requieren esencialmente de cuatro elementos [38]:

1. Un periodo de referencia. En la actualidad es la segunda quincena de julio del 2018.
2. Una canasta de bienes y servicios representativa de los hogares mexicanos. Esta se obtiene principalmente a partir de la Encuesta Nacional de Gastos de los Hogares (ENGASTO). Actualmente se cuenta con 299 genéricos, los cuales se encuentran distribuidos en 55 áreas geográficas del territorio mexicano.
3. Ponderadores para representar la contribución de gasto de los productos adquiridos por los consumidores. Esta es elaborada a partir de la ENGASTO, y la Encuesta Nacional de Ingresos y Gastos de los Hogares (ENIGH).
4. Un muestreo probabilístico para determinar los puntos de venta. Esto es con el fin de cotizar los precios de los bienes y servicios específicos.

Una vez que estos elementos se han establecido, se puede proceder al cálculo del INPC. Dicho proceso se obtiene en dos partes [38]:

PARTE 1. Se calculan los índices de precios elementales. Estos son índices de precios para los genéricos⁵ seleccionados (299), que abarcan las distintas áreas geográficas escogidas (55). Pueden estar clasificados como probabilísticos o no probabilísticos. En ambos casos se utiliza el índice de Jevons [38, 42]:

- **Índice elemental probabilístico:** En este caso se cuenta con 248 genéricos distribuidos en las 55 áreas geográficas.

Sea X un conjunto de n especificaciones de algún genérico G . Sea $x \in X$ una especificación, p_x^s el precio de la especificación x al tiempo s y p_x^t el precio de la

⁵Conjunto de bienes y servicios que comparten características similares entre sí (e.g. carnes: de chivo, de vaca, de pollo, etc.). Para este tipo de conjunto se tienen datos relacionados con el gasto y los ponderadores, los cuales se utilizan para el cálculo de índices de precios. Este conjunto pertenece a la categoría primaria o de agregado elemental [38].

especificación x al tiempo t . Entonces el índice elemental probabilístico, I_{ep} , se define como:

$$I_{ep} = \prod_{x \in X} \left(\frac{p_x^t}{p_x^s} \cdot 100 \right)^{\frac{1}{n}} \quad (2.6)$$

El I_{ep} denota un promedio geométrico en base a los radios $\frac{p_x^t}{p_x^s}$ de los precios de los n productos.

Observe que la ec. (2.6), puede ser de 3 tipos⁶ distintos. Esto depende del tipo de estrato⁷ con el que se cuente:

1. Con ponderadores que dependen del tipo de mercado (tradicional y moderno) y del tamaño del punto de venta.
2. Con ponderadores que solo dependen del tipo de mercado.
3. Con ponderadores independientes del tipo de mercado y del tamaño del punto de venta.

- **Índice elemental no probabilístico:** En este caso solo se cuenta con 51 genéricos distribuidos en las 55 áreas geográficas.

Sea X un conjunto de n especificaciones de algún genérico G . Sea $x \in X$ una especificación, p_x^R el precio relativo⁸ de la especificación x . Entonces el índice elemental no probabilístico, I_{enp} , se define como:

$$I_{enp} = \prod_{x \in X} \left(\frac{p_x^t}{p_x^s} \right)^{\frac{1}{n}} \quad (2.7)$$

PARTE 2. Se calculan los índices de precios de nivel superior. Son índices abstractos. Se utilizan los índices elementales para su cálculo. En esta categoría, se emplea el índice de Laspeyres [38]:

- **Índice de precios de nivel superior:** En este caso se involucran los 248 genéricos probabilísticos y los 51 genéricos no probabilísticos, dando un total de 299 genéricos. Todos dentro de las 55 áreas geográficas.

Sea X el conjunto que representa las 55 áreas geográficas escogidas para el cálculo de los índices de precios. Sea I_e un índice elemental que toma como valor al probabilístico I_{ep} o no probabilístico I_{enp} , y w_e el ponderador de gasto de los hogares correspondiente a I_e . Entonces el índice superior para el genérico x , $I_s(x)$, se define como:

$$I_s(x) = \sum_{x \in X} I_e w_e \quad (2.8)$$

Observe que la ec. (2.8) recoge cada índice de precios por genérico distribuido en las 55 áreas geográficas, para formar un sólo índice del genérico. Por ejemplo, suponga

⁶Dependiendo de la información con la que cuente el investigador [38].

⁷Conjunto de bienes o servicios agrupados en 3 categorías: por ubicación geográfica, tipo de mercado, tamaño de la unidad económica en base a sus ingresos por ventas, y que forman parte de una entidad mayor [38, 43].

⁸El precio relativo se define como $\frac{p_x^t}{p_x^s}$, $s, t \in \mathbb{R}$, y $s < t$, [42].

que se tiene un único genérico, el café. Asuma que el café solo se encuentra en Oaxaca, Cancún y Puebla. Entonces se calcula el índice de precios ponderado que involucre a los 3 estados, cada uno de ellos con su propio genérico del café, para crear un “índice general” $I_s(\text{café})$ tal que represente un “único” café.

Hasta aquí, ya se tiene toda la herramienta necesaria para definir el INPC o I_{Lpy} aplicado a los índices de precios de México.

Sea X el conjunto que representa los 299 genéricos. Sea $x \in X$ un genérico, $I_s(x)$ el índice del genérico x , y w_x el ponderador que representa el porcentaje de gasto de los hogares mexicanos, para el genérico x . Entonces, el INPC, se define como:

$$INPC = \sum_{x \in X} I_s(x)w_x \quad (2.9)$$

Note que la ec. (2.9) involucra directamente la ec. (2.8), pues se sustituye $I_s(x)$ en el INPC. Es decir, considera cada índice por cada genérico en todas las áreas geográficas. Esto conduce a formar un promedio general ponderado, para todos los índices de los distintos genéricos.

En resumen, se tiene una dependencia de índice tras índice tras índice. El INPC depende directamente de los índices superiores y los índices superiores dependen directamente de los índices elementales, los cuales a su vez pueden ser de 2 tipos.

Y es así como los economistas, estadísticos y matemáticos, del INEGI, obtiene el INPC de México.

Como en toda afirmación de carácter científico, en este caso es sobre economía-matemática, y también, debido a todos los trabajos que se realizan en base al cálculo del INPC, por diferentes investigadores, es necesaria la existencia de varios argumentos que proporcionen la efectividad del INPC. Así, se muestran los siguientes puntos tales que, al día de hoy, validan al INPC [38]:

- El INPC se ha estado elaborando desde el año de 1969. Esto ha traído como consecuencia tener suficiente experiencia para la elaboración del índice, al día de hoy.
- El índice tiene una periodicidad de cálculo y difusión frecuente. En el Diario Oficial de la Federación (DOF) se publica cada quincena y en <https://www.inegi.org.mx/> cada quincena y cada mes.
- El INPC posee la certificación de calidad Organización Internacional para la Estandarización (ISO) 9001 de los procesos de elaboración de los índices, desde el 2011.
- El cálculo del índice considera todas las localidades urbanas del país. A la vez toma al menos una ciudad de un estado, para que lo represente.
- En el índice se toman en cuenta los bienes y servicios adquiridos por consumidores rurales y urbanos.
- El INPC considera distintos puntos de venta para cotizar los bienes y servicios.

Para los puntos de ventas, se basa en muestreos probabilísticos y en las preferencias de los consumidores.

- El índice incluye variedades para los distintos tipos de bienes y servicios (marca, calidad y presentación).
- La canasta para el INPC se forma, principalmente, a través de una encuesta realizada por el INEGI a los consumidores. La encuesta se llama ENGASTO. En caso de detección de incongruencias en los datos, se consultan otras fuentes de información. Por ejemplo: la ENIGH 2014, el Sistema de Cuentas Nacionales de México (SCNM), los Censos Económicos (CE), la Encuesta Nacional de Adicciones (ENA), Estadísticas de producción y comercio, encuestas externas, registros administrativos, entre otros.
- Los ponderadores del INPC se actualizan de acuerdo al proceso de Ciclo de Actualización Económica⁹.

Dicho lo anterior, después de haber realizado una revisión del tema, es conveniente ver ejemplos que ilustren el papel del INPC, con el fin de percibirlo de manera más táctil. A continuación se muestran algunos ejemplos donde el INPC sirve de guía para la toma de decisiones [5, 25]:

- Familias
 - Para el presupuesto de consumo y ahorro.
- Empresas
 - Para cuestiones de capital, inversión y producción.
- Gobierno
 - Para la elaboración o modificación de políticas fiscales y monetarias, en el incremento de los salarios, montos de jubilación y prestaciones del seguro social.

Como punto final de la sección y después de tener el conocimiento antes mencionado, es natural preguntarse ¿quién es la autoridad responsable de recopilar, procesar y divulgar el INPC, y por qué?, dicha tarea le corresponde al INEGI y tal atributo se establece en el Código Fiscal de la Federación (CFF) en el segundo párrafo del artículo 20 [5].

Como observación final se hace la aclaración de que en términos económicos existen diferentes índices para indicar los cambios en los precios, por ejemplo¹⁰, por ejemplo, índice: de Fisher, Walsh, Young, Törnqvist-Theil, Lloyd-Moulton, Konüs, Carli, Rothwell, [42], de precios al consumidor, de precios al consumidor armonizado, de precios industriales, etc.

⁹“Consiste en un proceso de actualización integral y sistemático de todos los programas que generan Información Estadística de carácter económico, que permite garantizar una mayor coherencia entre todas las fuentes y programas con el fin de lograr una mayor fortaleza técnica, y del mismo modo en la Información que se genera y proporciona, destacando lo indicado en cada uno de los presentes temas, a fin de mantener las estadísticas, indicadores y mediciones macroeconómicas, con la calidad, pertinencia, veracidad y oportunidad y bajo los principios de accesibilidad, transparencia, objetividad e independencia requeridas por el Sistema Nacional de Información Estadística y Geográfica.” [44]

¹⁰Algunos de ellos son resultado de algún tipo de combinación entre otros índices.

2.4. Efectos de la inflación

La política económica en México tiene como objetivo estimular el crecimiento del PIB, la demanda agregada, el consumo, la inversión, así como disminuir la tasa de desempleo. Por lo cual es necesario establecer ideas, planes y acciones en pro del éxito macroeconómico. Una prioridad dentro de la política económica es mantener un control adecuado de la inflación, porque los efectos negativos afectan a los agentes económicos: las familias, empresas y el gobierno. Así, el tema de la inflación es relevante en el ámbito de la economía de un país.

De forma más precisa, y de acuerdo a la definición de inflación en la sección 2.1, la inflación afecta el poder adquisitivo, y por tanto el bienestar de los agentes económicos; por ejemplo, suponga un país hipotético en el cual el precio de los boletos de avión aumenta 10 veces más, durante el periodo de 1 mes, y los distintos salarios de los turistas se mantienen fijos, durante este tiempo. Entonces es probable que exista algún turista que ya no podrá visitar cierto lugar, al menos de forma temporal. Así, el turista tendrá que optar por un bien sustituto (que no pertenezca a la canasta representativa), o asimilar la posible frustración.

Ahora, dentro de la teoría económica se tiene un marco referencial para el análisis de la inflación y sus efectos, por ejemplo la teoría cuantitativa del dinero y la curva de Phillips. Así también se tienen instrumentos como la política fiscal y la política monetaria que influyen en la demanda agregada, la inflación, la oferta monetaria, la tasa de interés, el nivel de empleo, entre otras variables macroeconómicas; la aplicación de éstos instrumentos depende de la fase del ciclo económico en que se encuentre la economía y de los objetivos de política económica que se tengan como metas [45], [46].

2.5. Métodos de pronóstico para variables económicas

Los métodos de pronóstico, estadísticos, matemáticos o computacionales, aplicados a una determinada área, son aquellos procedimientos, a partir de los cuales se puede generar conocimiento nuevo, con bases sólidas y con un margen de error manipulable. Una vez que se ejecuta el método y se interpreta de forma eficiente, éste tiene el poder de guiar a los distintos agentes económicos: familias, empresas y gobierno. La guía se da a través de las expectativas que ofrece el pronóstico. Los pronósticos son herramienta de gran ayuda para los agentes, pues devuelven una visión futura sobre algún fenómeno. Es así como ayuda a los agentes a tomar decisiones, ya que éstas son premeditadas y por tanto más acertadas.

A continuación se presentan trabajos que muestran métodos de pronóstico estadísticos, computacionales y matemáticos, que más se han aplicado dentro del área de economía:

En [47] se utilizaron redes neuronales artificiales como alternativas para analizar la serie del Índice de Precios al Consumidor (IPC) para Ecuador, en el periodo de 2000 al 2017. Esto con el fin de dar su respectivo pronóstico. Los resultados se contrastaron con los resultados de modelos estadísticos tradicionales: Movimiento Promedio Integrado Autorregresivo (ARIMA) y Vector Autorregresivo (VAR). Para analizar la calidad de los pronósticos hechos, se realizó un pronóstico recurrente. Dicho pronóstico recurrente indicó que los tres modelos eran estables. Otro artículo que muestra la eficiencia de las redes neuronales aplicadas al IPC es [48]. Aquí se realizó una predicción que muestra los resultados mensuales del IPC, aplicando una RNA borrosa. Los resultados indicaron un buen ajuste del pronóstico en comparación con la serie real.

En lo que respecta a la red neuronal perceptrón multicapa, en [49] se encontró que fue el modelo más eficiente, en el pronóstico de precios de codominios. Dicho modelo neuronal se comparó, utilizando la Raíz del Error Cuadrático Medio (RECM), con técnicas estadísticas tradicionales. Así también, otro artículo en el que el perceptrón multicapa fue el método más destacado es en [50]. Aquí se comparan 4 métodos distintos: Máquinas de Vectores de Soporte (SVM), Análisis de Regresión Múltiple (ARM), ARIMA y RNA (perceptrón multicapa), para el pronóstico de la demanda de turismo en Indonesia. Se encontró que, de acuerdo al ECM y al porcentaje de precisión, el mejor modelo fue el perceptrón multicapa.

De esta forma, con los resultados observados en los distintos artículos, se obtiene la impresión de que las redes neuronales artificiales son eficaces para la elaboración de pronósticos; lo cual refuerza la elección de una RNA, en este trabajo. A su vez, se aclara que las medidas de eficiencia para las redes que se consideran en tales artículos no necesariamente serán las que se usen para calificar el pronóstico de esta investigación.

En el presente capítulo se estudió lo referente a la inflación por medio del INPC. En el siguiente capítulo se construirá una RNA perceptrón multicapa, la cual propiciará el pronóstico para el INPC o bien, pronóstico de la inflación en México.

CAPÍTULO 3

Aplicación de la RNA para el pronóstico de la inflación

En este capítulo se lleva a cabo la aplicación¹ de una red neuronal artificial, perceptrón multicapa, para la estimación del pronóstico del INPC subyacente en un marco de tiempo de mediano plazo. Se comienza con la selección de los datos que se introducirán a la red, luego se prosigue con la elección de la red perceptrón multicapa, después se describe lo referente al entrenamiento con un aprendizaje supervisado mediante el algoritmo de propagación hacia atrás. Al finalizar la parte del entrenamiento, se analiza lo concerniente al pronóstico con el objetivo de elegir la mejor estructura de red, tal que pueda realizar aproximaciones eficientes. Por último se finaliza con la conclusión sobre la hipótesis que guía la presente investigación.

3.1. Datos históricos seleccionados para alimentar la red

Los datos seleccionados para introducir a la red, corresponden al INPC subyacente, con el intervalo de enero de 1982 a septiembre de 2021. Véase figura 3.1. La información del INPC fue recuperada de INEGI, y los datos explícitos se muestran en el apéndice A; y el año base es la segunda quincena de julio de 2018 [51]. La muestra se dividió en dos periodos con el fin de elaborar 2 predicciones. Con el primer pronóstico se entrena y válida la red obteniendo la estructura óptima para el pronóstico y con la otra proyección se conocerá el INPC a tiempo futuro fuera de la muestra:

De enero de 1982 a marzo de 2020, corresponde a la muestra que se le denominó TA. TA fue el conjunto para la realización del pronóstico A. Se pronostican 18 datos mensuales que corresponden a abril del 2020 a septiembre del 2021.

De enero de 1982 a septiembre de 2021 corresponde a la muestra, que se le denominó TB. TB fue el conjunto para la realización del pronóstico B. Se pronostican 18 datos mensuales que corresponden a octubre del 2021 a marzo del 2023. Se entiende que cuando se haga referencia pronóstico B, se estará refiriendo a la predicción a medio plazo, que se genera después del intervalo de tiempo de TB.

Así, TA y TB, son los datos que se introducirán en el modelo neuronal específico

¹Por aplicación se define a la evaluación o verificación de la RNA, a través de la introducción o unión de un conjunto de datos, con ésta.

En adelante se utilizan los siguientes términos como sinónimos: red neuronal artificial, RNA, red, red neuronal, modelo, modelo neuronal, y modelo neuronal artificial.

que se elegirá.

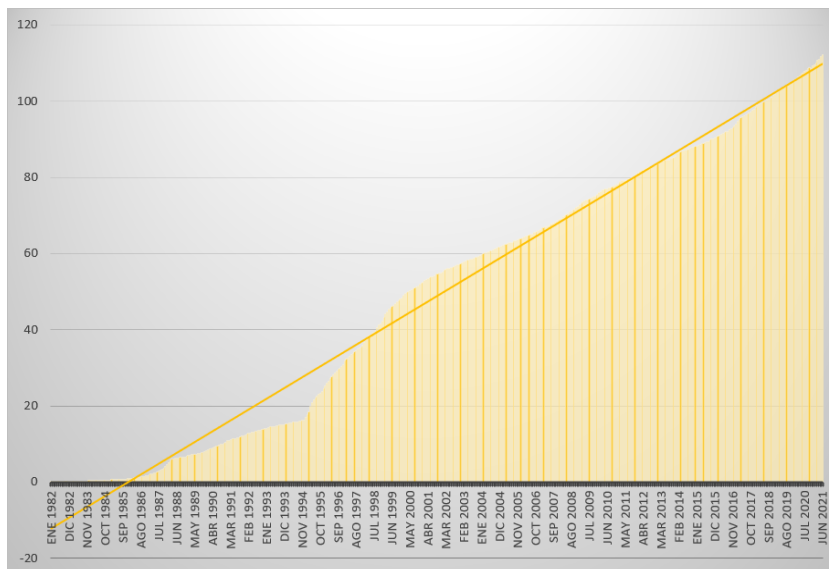


FIGURA 3.1. Índice Nacional de Precios al Consumidor: Subyacente.

Nota: Base segunda quincena de Julio 2018 (mensual).

Fuente: Elaboración propia con datos del INEGI.

3.2. Diseño del tipo de red neuronal

Como variable económica a pronosticar en este trabajo de investigación se seleccionó el INPC con un intervalo de tiempo de enero de 1982 a septiembre de 2021, esta serie histórica de datos representa una serie de tiempo.

De los diferentes tipos de diseño de una red neuronal que se mencionan en la literatura provista en el software de Matlab, se identificó que una RNA no lineal autorregresiva de propagación hacia adelante (NAR), está diseñada para predecir valores de una serie de tiempo, es decir, a partir de conocer los valores pasados de dicha serie $y(t - n), y(t - n + 1), \dots, y(t)$, pronostica $y(t + 1), y(t + 2), \dots$, [52].

Dadas las características de una NAR, en la presente investigación se considera el diseño de esta red para realizar el pronóstico del INPC. La red NAR es una RNA de propagación hacia adelante, sigue la estructura de un perceptrón multicapa que considera una capa de entrada, una o más capas ocultas y una capa de salida. Véase 3.2.

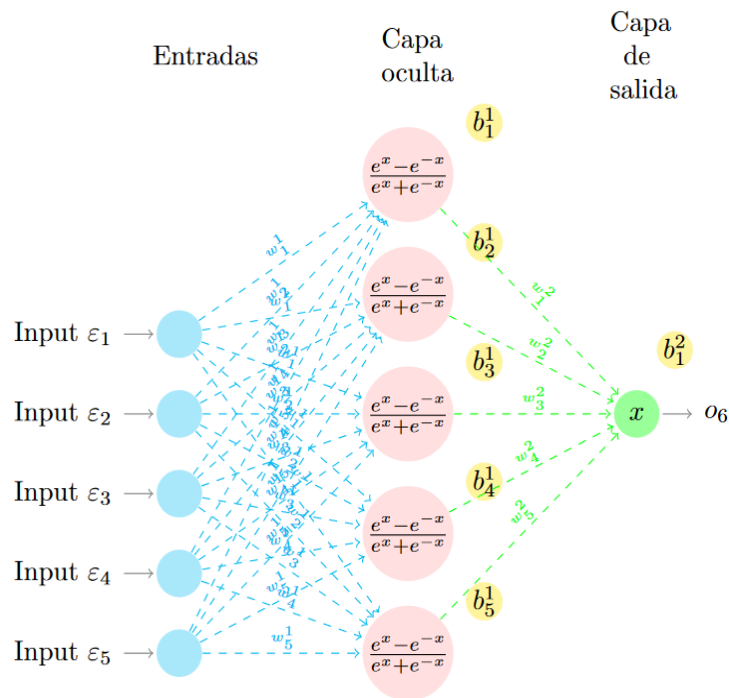


FIGURA 3.2. Representación de la Red NAR, con la Función Sigmoide y Lineal en sus Capas.

Fuente: Elaboración propia en base a [53].

A su vez, este tipo de RNA utiliza, la función tangente hiperbólica sigmoide y función lineal en su(s) capa(s) oculta(s), y capa de salida, respectivamente. Se eligen o confirman estas dos funciones de activación debido a que la tangente hiperbólica sigmoide acota a un rango numérico limitado: $[-1, 1]$, y tiene simetría asintótica [54] (por ejemplo una consecuencia de esto es que, en general, bajo estas condiciones es más probable que los valores de salida de la red estén entre $[-1, 1]$); y la elección de la función lineal es porque se está resolviendo un problema de regresión, así, se requiere del uso de una función lineal en la última capa [55].

En la siguiente sección se aborda lo respectivo al entrenamiento, la selección de la estructura de la red, y el pronóstico A sobre el INPC.

3.3. Entrenamiento de la red NAR

Al elegir una RNA tipo NAR, con una capa de entrada, una capa oculta y una de salida, se inicia el entrenamiento con la muestra de datos TA, que considera el periodo de enero de 1982 a marzo de 2020, conformada por 459² valores mensuales

²Se consideran 459 datos debido a que son el total de datos mensuales dentro del periodo de enero de 1982 a marzo de 2020, es decir, se aparta el resto de los 18 datos de la base de datos, con el fin de realizar y verificar el primer pronóstico; a su vez, se tomaron los 459 valores debido a que se quería obtener un panorama lo más completo posible, para así conocer el comportamiento de la serie de forma más acertada. (Anteriormente se tenía el número 477, mi error. También se modificaron, en las sig. hojas, los datos del INPC a 459, 458, etc.)

del INPC subyacente, que representan los datos con los que se alimenta la red y a su vez se vuelven los datos objetivo con los que se compara la salida de la red.

TABLA 3.1. Muestra de Datos TA.

Número de dato	Valor que corresponde de INPC
1	0.092900756
2	0.096868689
3	0.101013577
⋮	⋮
457	105.5770987249
458	105.9614303519
459	106.272453486

Fuente: Elaboración propia con datos del INEGI.

Matemáticamente, los valores que corresponden al INPC se pueden expresar como un vector de entrada de la manera siguiente.

$$\mathbf{p} = \begin{pmatrix} 0.092900756 \\ 0.096868689 \\ 0.101013577 \\ \vdots \\ 105.5770987249 \\ 105.9614303519 \\ 106.272453486 \end{pmatrix}$$

Luego se define la matriz \mathbf{W}^1 . Como condición inicial estos pesos se establecen de manera aleatoria con valores que se encuentren en el intervalo $[-2,2]$ [56, 57]; \mathbf{W}^1 está definida explícitamente como [4]:

$$\mathbf{W}^1 = \begin{pmatrix} w_{1,1}^1 & \cdots & w_{1,459}^1 \\ w_{2,1}^1 & \cdots & w_{2,459}^1 \end{pmatrix} \rightarrow \begin{pmatrix} .01 & \cdots & .1 \\ .02 & \cdots & .2 \end{pmatrix}$$

Luego, el vector de entradas \mathbf{p} se multiplica por la matriz de pesos \mathbf{W}^1 , y así, son introducidos a la capa de la red [4]:

$$\begin{pmatrix} .01 & \cdots & .1 \\ .02 & \cdots & .2 \end{pmatrix} \cdot \begin{pmatrix} 0.092900756 \\ 0.096868689 \\ 0.101013577 \\ \vdots \\ 105.5770987249 \\ 105.9614303519 \\ 106.272453486 \end{pmatrix}$$

Después se ejecuta el operador \sum definido por $\mathbf{W}^1\mathbf{p} + \mathbf{b}^1$, donde \mathbf{b}^1 representa el vector bias de las neuronas en la capa oculta. Como condición inicial al igual que los pesos, se definen de manera aleatoria pero dentro del intervalo $[-1, 1]$ [56, 57]. Dado que de manera inicial solo se están suponiendo 2 neuronas en la capa oculta, entonces

únicamente se consideran 2 bias [4]:

$$\mathbf{b}^1 = \begin{pmatrix} .03 \\ .04 \end{pmatrix}$$

Se tiene entonces:

$$\sum = \mathbf{W}^1 \mathbf{p} + \mathbf{b}^1 = \begin{pmatrix} (.01)(0.092900756) + \dots + (.1)(106.272453486) + .03 \\ (.02)(0.092900756) + \dots + (.2)(106.272453486) + .04 \end{pmatrix}$$

El resultado de haber aplicado el operador \sum es un vector denominado vector neto, \mathbf{n}^1 . A continuación, \mathbf{n}^1 se traslada al vector \mathbf{f}^1 de funciones de activación tangente hiperbólica sigmoide para ser evaluado [58, 59, 4]:

$$\begin{aligned} \mathbf{f}^1(\mathbf{n}^1) &= \begin{pmatrix} \frac{e^{n_1} - e^{-n_1}}{e^{n_1} + e^{-n_1}} \\ \frac{e^{n_2} - e^{-n_2}}{e^{n_2} + e^{-n_2}} \end{pmatrix} \\ &= \begin{pmatrix} \frac{2}{1 + e^{-2n_1}} - 1 \\ \frac{2}{1 + e^{-2n_2}} - 1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{2}{1 + e^{-2((.01)(0.092900756) + \dots + (.1)(106.272453486) + .03)}} - 1 \\ \frac{2}{1 + e^{-2((.02)(0.092900756) + \dots + (.2)(106.272453486) + .04)}} - 1 \end{pmatrix} \end{aligned}$$

El resultado de la evaluación o cálculo se determinó que se encuentra dentro del intervalo $[-1, 1]$ [59, 58].

A tal resultado de la evaluación, se le etiqueta con el nombre del vector \mathbf{a}^1 . Este vector \mathbf{a}^1 resulta ser la entrada para la tercera capa. El proceso de entrada y demás ejecuciones en esta tercera capa es análogo al proceso que se llevó a cabo en la capa oculta [4].

Dado que la última capa tiene la función de activación lineal, $\mathbf{f}^2(\mathbf{n}^2) = \mathbf{n}^2$, entonces $\mathbf{f}^2(\mathbf{n}^2)$ devuelve el mismo resultado, que se denota como \mathbf{a}^2 . Tal \mathbf{a}^2 representa la salida de la red. Matemáticamente se puede expresar por medio de la composición de funciones [4]:

$$\begin{aligned} \mathbf{a}^2 &= f^2(\mathbf{n}^2) \\ &= f^2(W^2 \mathbf{a}^1 + b^2) \\ &= f^2(W^2 f^1(\mathbf{n}^1) + b^2) \\ &= f^2(W^2 f^1(W^1 \mathbf{p} + b^1) + b^2) \\ &= W^2 f^1(W^1 \mathbf{p} + b^1) + b^2 \end{aligned}$$

La figura 3.3 define la condición inicial del perceptrón multicapa, de la cual se parte para iniciar el entrenamiento y obtener la red óptima que permita pronosticar

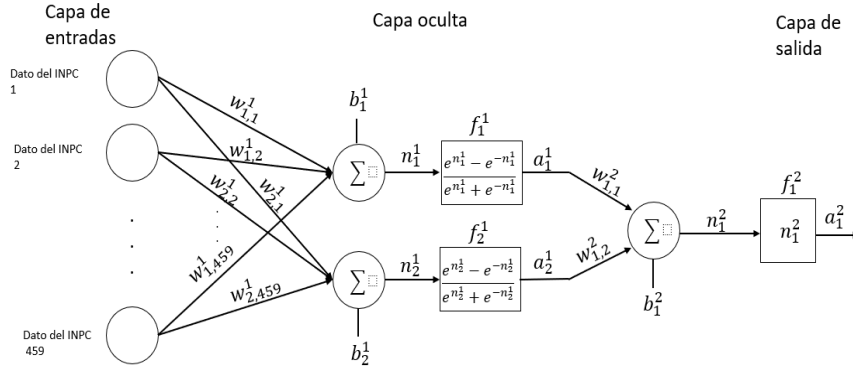


FIGURA 3.3. Perceptrón Multicapa Condición Inicial.
Fuente: Elaboración propia.

el INPC, con un error mínimo, cuyo valor puede caer en el intervalo $(-.01, .01)$, tal como se establece en la hipótesis de investigación.

En cuanto al inicio del entrenamiento, éste se obtuvo a partir de la primer iteración $k = 1$; se comparó a_1^2 con el primer valor original del INPC ta_1 , obteniendo una diferencia cuyo valor se alejaba de 0, por lo tanto fue necesario realizar otra iteración, y así sucesivamente hasta minimizar el error mediante un proceso de aproximación al utilizar el algoritmo de propagación hacia atrás.

La aplicación del algoritmo se describe a continuación. Con el fin de iniciar la aproximación de la salida de la red a_1^2 con el dato objetivo ta_1 y minimizar el nivel de error, fue necesario el cálculo del valor de la sensibilidad que indica en que medida deben de modificarse los parámetros, como son los pesos y bias. Matemáticamente, en primer lugar se obtuvo la sensibilidad de la capa de salida mediante la derivada parcial de la función de activación correspondiente, dicha sensibilidad se representa con el valor de s^2 [4]. De forma explícita [4]:

$$s^M = -2 \frac{\partial \mathbf{F}^M(\mathbf{n}^M)}{\partial \mathbf{n}^M} (\mathbf{t} - \mathbf{o}) = -2 \frac{\partial \mathbf{F}^2(\mathbf{n}^2)}{\partial \mathbf{n}^2} (\mathbf{t} - \mathbf{o}) = -2(1)(1) = -2 = s^2$$

Una vez evaluado s^2 , éste se sustituye en la ecuación correspondiente a s^1 , es decir, la sensibilidad se propaga hacia atrás [4]:

$$\begin{aligned} s^1 &= \frac{\partial \mathbf{F}^m(\mathbf{n}^m)}{\partial \mathbf{n}^m} (\mathbf{W}^{m+1})^T s^{m+1} \\ &= \frac{\partial \mathbf{F}^1(\mathbf{n}^1)}{\partial \mathbf{n}^1} (\mathbf{W}^2)^T s^2 \\ &= \begin{pmatrix} \frac{4e^{-2n_1}}{(1+e^{-2n_1})^2} & 0 \\ 0 & \frac{4e^{-2n_2}}{(1+e^{-2n_2})^2} \end{pmatrix} \begin{pmatrix} .01 & \dots & .1 \\ .02 & \dots & .2 \end{pmatrix} (-2) \end{aligned}$$

Después, s^2 y s^1 se sustituyó en (3.1), (3.2), y en (3.3), (3.4), respectivamente,

con el fin de actualizar los pesos y bias correspondientes a la capa de salida y a la capa oculta, respectivamente [4, 10]:

$$\mathbf{W}^m(k+1) - \mathbf{W}^m(k) = -\alpha s^2 (\mathbf{o}^{m-1})^T \quad (3.1)$$

$$\mathbf{b}^m(k+1) - \mathbf{b}^m(k) = -\alpha s^2 \quad (3.2)$$

y

$$\mathbf{W}^m(k+1) - \mathbf{W}^m(k) = -\alpha s^1 (\mathbf{o}^{m-1})^T \quad (3.3)$$

$$\mathbf{b}^m(k+1) - \mathbf{b}^m(k) = -\alpha s^1 \quad (3.4)$$

Con esto, se actualizaron los parámetros de pesos y bias, de las ecuaciones (3.1), (3.2), (3.3), y (3.4), respectivamente. Al mismo tiempo se consideró una tasa de aprendizaje α de .10, cabe señalar que la misma puede tomar valores que pertenece al intervalo (.01, .99) mencionado en [18].

Una vez que se actualizaron los pesos y los bias con este proceso de propagación hacia atrás de las sensibilidades, se estimó de nuevo la salida de la red y se observó que continuaba presentándose un nivel de error, como por ejemplo, supóngase que $a_1^2 = 4.2$ y $ta_1 = 3.2$, entonces el error es $e = a_1^2 - ta_1 = 4.2 - 3.2 = 1$. Por lo tanto fue necesario volver a repetir el proceso de propagación hacia atrás de las sensibilidades calculadas, para ajustar los pesos hasta que estos fueron óptimos y se logró minimizar el error de entrenamiento [4].

El proceso anterior fue una descripción general, de la forma en que opera el algoritmo de propagación hacia atrás.

Un concepto que se omitió en esta descripción es el término de rezago, el cual va a ser explicado a mayor profundidad en el siguiente apartado, dada su importancia en el diseño de la estructura óptima de la red que permita minimizar el ECM del pronóstico.

3.4. Selección óptima de la estructura de la red NAR para el primer pronóstico con base en TA

En la sesión anterior se detalló de manera general el entrenamiento de la red mediante el cual se minimiza el error entre las entradas de la red y las salidas, a través del proceso de aproximación utilizando el algoritmo propagación hacia atrás. Ese fue el primer ejercicio que se realizó en el presente trabajo de investigación. En la siguiente etapa, para obtener el pronóstico sobre el INPC, es necesario incorporar a la fase de entrenamiento el concepto de rezago y de estructura, así como modificar el número de neuronas ocultas, coef. de aprendizaje y rezagos, mediante una serie de iteraciones con base a prueba y error; el objetivo ahora es minimizar el ECM del pronóstico.

Los rezagos o variables con rezagos, son considerados en este modelo neuronal artificial, debido a que la variable de estudio se encuentra en el ámbito económico, en el cual se suele trabajar con series de tiempo. Al realizar un pronóstico es importante analizar los datos históricos de dicha variable, que influyen en el comportamiento futuro de la misma dado que existe una dinámica en la que toda causa económica

produce efectos durante un periodo de tiempo; esto es, no solo se genera una afectación puntual [60]. Tales rezagos, se utilizan para poder generar nueva información, uno a uno, relativa al tema [61, 62].

Dicho lo anterior, ahora ya se puede definir una estructura. La estructura de la RNA consiste de una sección de rezagos para la capa de entrada, un conjunto de neuronas para la capa oculta, parámetros (pesos y bias) para cada capa, y de un coeficiente de aprendizaje α .

Con una estructura arbitraria y la muestra TA, ya se puede proceder con las ejecuciones del script en MATLAB.

En seguida se muestra el código, C, que se elaboró para el entrenamiento y pronóstico de la red. Éste fue modificado a partir del script que se muestra en el apéndice B.0, el cual se generó por medio de la aplicación Neural Net Time Series, de MATLAB. Con base a los aspectos teóricos de las redes, se realizaron numerosas pruebas con distintos códigos, a través de distintos modelos neuronales artificiales, para finalmente obtener el código C que servirá de base para el pronóstico del INPC.

Nombre del script: código C para el pronóstico A

```
t=xlsread('/Users/VeronicaRM/INPCoct_28.xlsx','Sheet1','d4:d462');
%t deja los ultimos 18 datos, porque con esos se medira el ECM.
%t contiene todos los datos a considerar para generar el pronostico
%de 18 datos
T = tonndata(t,false,false);

tt=xlsread('/Users/VeronicaRM/INPCoct_28.xlsx','Sheet1','d463:d480');
%este elemento de arriba contiene los ultimos 18 datos
TT = tonndata(tt,false,false);

trainFcn = 'trainlm';

num=70;
feedbackDelays = 1:num;
hiddenLayerSize = 15;

net = narnet(feedbackDelays,hiddenLayerSize,'open',trainFcn);

[x,xi,ai,tn] = preparets(net,{}, {},T);

net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

net.trainParam.lr=1;
%net.divideMode='value';

%Valor minimo del gradiente
net.trainParam.min_grad=1e-20;
```

```

%For validation checks (maximo numero):
net.trainParam.max_fail=12;

%.6 es mejor a .1, definitivamente
net.trainParam.epochs=50000;

[net,tr] = train(net,x,tn,xi,ai);

%.....
%...PRONOSTICO.....
%A
nets = removedelay(net);
[xs,xis,ais,ts] = preparets(nets,{}, {},T);
pro1=nets(xs,xis,ais);
TamPro1ConUnPrimerPronostico=size(pro1);

V=T;

xa=size(t);
xb=xa-num;

for i=1:17
    V(xa(1)+i)=pro1(xb(1)+i);
    [xs2,xis2,ais2,ts2] = preparets(nets, {}, {},V);
    pro1=nets(xs2,xis2,ais2);
end
b=cell2mat(pro1(end-17:end));
bb=reshape(b,18,1);
display(bb,'18 datos pronosticados')

%.....

%....END PRONOSTICO.....
%.....

    e18=gsubtract(TT,pro1(end-17:end));
    ECM=mse(e18);
    display(ECM,'ECM de los 18 datos')

%.....
%...WEIGHTS & BIAS.....
W1=net.IW{1};
display(W1,'Pesos: de los inputs a la primer capa')

Wb=net.LW{2};
display(Wb,'Pesos: de la primer capa a la capa de salida')

b1=net.b{1};
display(b1,'Bias de la primer capa oculta')
b2=net.b{2};
display(b2,'Bias de la capa de salida')

```

```

%.....
%....END WEIGHTS & BIAS.....

%.....
%.....LOS WHILES.....

%Para la generación de datos de la tabla 3.3:
%Poner iter=1; minimo=20000; OCTOBER_inpc (en command windows)

% while(ECM>.01 && iter<20)
%     iter=iter+1;
%     if ECM<minimo
%         minimo=ECM;
%     end
%     OCTOBER_inpc;
% end

% if(ECM <minimo)
%     minimo=ECM;
%     display(minimo)
% else
%     display(minimo)
% end

%Para conseguir cumplir la hipótesis, dentro de un límite
%de 5,000 ejecuciones, y contar el núm.
%de iteraciones que necesito:
%Poner iter=1; minimo=1; set(0,'RecursionLimit',5000); OCTOBER_inpc
%(en command windows)

while(ECM>.01)
    iter=iter+1;
    if ECM<minimo
        minimo=ECM;
    end
    OCTOBER_inpc;
end
minimo=ECM;
display(minimo)
%.....END LOS WHILES.....
%.....

```

De acuerdo a la teoría de redes neuronales artificiales, no hay una metodología precisa que resulte en la elección del número de capas ocultas, número de neuronas en cada capa, y el coeficiente de aprendizaje alpha, [4, 18, 10]. La manera de proceder para encontrar los elementos óptimos de la RNA que contribuyan a que la red arroje un ECM mínimo de la proyección, es mediante repetidas iteraciones del código C, usando ciclos, con distinto número de capas, neuronas, y coeficientes de aprendizaje [4, 18]; a su vez, el mismo método aplica para la determinación del número de rezagos. En cuanto a la determinación de los parámetros de pesos y bias óptimos, estos quedan definidos a través del algoritmo de propagación hacia atrás [4], que a su vez se registrará

a partir de haber elegido el ECM mínimo del pronóstico A.

Para ejemplificar, la forma en que se encuentra la estructura óptima de la red mediante la cual se obtiene el ECM mínimo del pronóstico A, en la tabla 3.2 se muestra como ejemplo la estructura de 10 redes distintas, elegidas dentro de las 42 propuestas de estructuras, que se muestran en la tabla 3.12. Cada estructura fue introducida en el código C, y generó su pronóstico respectivo (véase figura 3.4).

TABLA 3.2. Diferentes estructuras para la RNA.

	Rezagos	Neuronas	α
ERN1	50	20	1.75
ERN2	60	20	1.75
ERN3	70	10	1
ERN4	80	20	1
ERN5	60	25	1
ERN6	40	25	2
ERN7	50	20	1.5
ERN8	60	30	1.5
ERN9	50	25	2
ERN10	60	15	1.5

Fuente: Elaboración propia con datos del INEGI y de la RNA.

Nota: ERN significa estructura de la red neuronal.

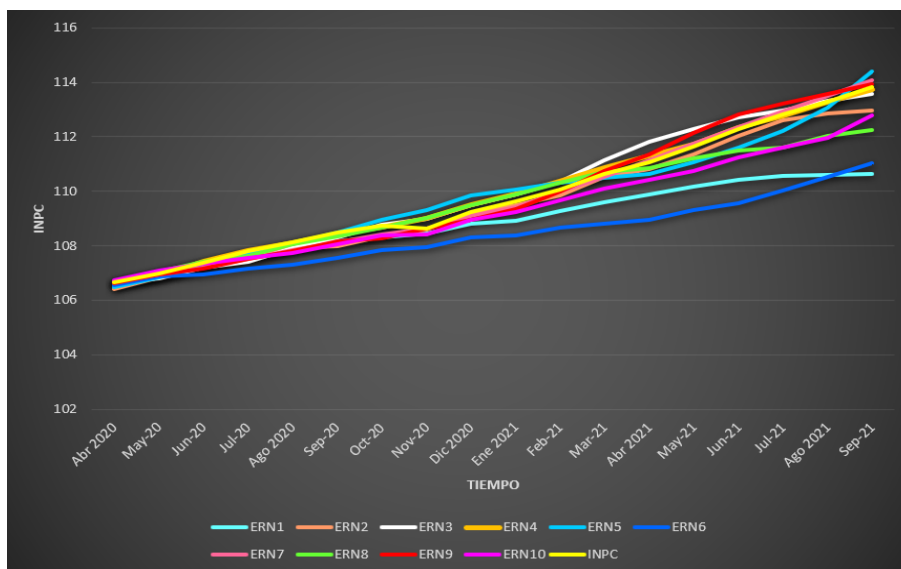


FIGURA 3.4. Pronósticos obtenidos con diferentes estructuras de la RNA.

Fuente: Elaboración propia con datos del INEGI y de la RNA.

Los múltiples pronósticos que se muestran en la figura 3.4 tienen asociado un ECM que va modificándose de acuerdo se modifique el número de neuronas en la capa oculta, la cantidad de rezagos y el coeficiente de aprendizaje. Cada iteración en la que se modifiquen dichos parámetros se denomina como ciclo while que es una instrucción en Matlab para llevar a cabo una o más tareas es utilizado para ejecutar repetidamente los scripts, de forma unitaria, estas repeticiones dentro del ciclo while, aplican hasta que se llega al límite numérico establecido para el ECM.

En las pruebas de ensayo y error llevadas a cabo³ mediante este ciclo while se consideraron en promedio un número cercano a 5,000 iteraciones con distintas combinaciones para la estructura de la NAR, el objetivo definir la estructura de la red que minimice el ECM del pronóstico A en la tabla 3.3 se muestra el ECM obtenido por cada combinación de parámetros (rezagos, neuronas en la capa oculta, y coeficiente de aprendizaje).

TABLA 3.3. ECM de los 18 Datos Aproximados, a Partir de Diferentes Estructuras de la Red, del Pronóstico A.

		hasta 20 o hasta que ECM < .01 entre 1 y 5000							
		1:50		1:60		1:70		1:80	
		Coeficientes de aprendizaje							
		1	1	0.5	1	0.5	1	0.5	
Neuronas	25	0.0203	0.0098	0.0099	0.0046	0.0055	0.0047	0.0038	E
en	30	0.0132	0.0122	0.0043	0.0126	0.0155	0.0082	0.0049	C
la	20	0.0115	0.006	0.0069	0.0162	0.0109	0.0074	0.0099	M
capa	10	0.0137	0.0217	0.008	0.0065	0.0095	0.0091	0.0146	
oculta	15	0.0148	0.0088	0.0065	.0033*	0.0238	0.0127	0.0098	
	22	0.0179	0.0089	0.0085	0.0037	0.0074	0.0124	0.0091	

Fuente: Elaboración propia.

La tabla 3.3 muestra que la estructura de la red NAR que minimiza el ECM para el pronóstico A es la que tiene 15 neuronas en la capa oculta con 70 rezagos, y $\alpha = 1$, con un ECM de .0033.

Una vez elegida la mejor estructura, también se conocieron los parámetros óptimos, las matrices de pesos \mathbf{W}^m y los vectores de bias \mathbf{b}^m , donde $m = 1, 2$. Los factores se muestran a continuación:

³Cabe señalar que en el proceso para identificar la estructura óptima para realizar el pronóstico del INPC se realizaron una serie de ejercicios los cuales partieron de un código inicial fundamentado en el software de Matlab en la caja de herramientas de redes neuronales, el cual represento una condición inicial con valores aleatorios respecto a los bias y pesos de la red y n números de neuronas en la capa oculta. A esta condición inicial se le fueron cambiando los parámetros incluidos el número de rezagos, el objetivo fue identificar la estructura con un ECM mínimo. Este ejercicio de ensayo de prueba y error requirió un trabajo extenso de programación computacional durante un tiempo aproximado de cinco meses de manera continua. En la tabla 3.3 de alguna manera resume todo el trabajo realizado.

TABLA 3.4. Vector de Bias de la Primer Capa de la Red.

B1
1.43
1.315
0.8387
-0.6698
-0.5398
-0.603
-0.2861
0.2117
0.1974
0.2326
-0.945
-1.1514
1.2351
1.4273
-0.6142

Fuente: Elaboración propia.

TABLA 3.5. El Bias de la Neurona de la Última Capa.

b2
0.2987

Fuente: Elaboración propia.

TABLA 3.6. Columna de Pesos de la Primer Capa a la Última de la Red.

P_2
-0.6998
-0.0299
-0.7662
-0.8528
0.3586
0.5589
-1.0256
-0.5367
-0.0647
-0.6425
-0.8989
0.7377
0.2607
0.1924
-0.6492

Fuente: Elaboración propia.

TABLA 3.7. Primeros y Últimos 3 Pesos, de la Matriz, de la Capa de Entrada a la Capa Oculta de la Red (en Forma de Columna).

P_1
-0.1895
-1.0959
-0.5948
⋮
-0.4227
0.089
0.0003

Fuente: Elaboración propia.

Las tablas 3.8 y 3.11 resumen las características generales de la RNA para realizar el pronóstico A y B, respectivamente.

TABLA 3.8. Características Generales de la RNA para Realizar el Pronóstico A.

Periodo de entrenamiento, validación y prueba	enero 1982 – marzo 2020
Periodo de pronóstico	abril 2020 – septiembre 2021
Estructura de la RNA	1 capa oculta con 15 neuronas, 1 capa de salida con 1 neurona; con las funciones de activación tangente hiperbólica sigmoide y lineal, respectivamente
Coef. de aprendizaje	1
Rezagos	70
Límite máximo de la verificación de validación	12
Límite mínimo del gradiente	1.00e-20

Fuente: Elaboración propia.

TABLA 3.9. Error Cuadrático Medio Observado.

ECM arbitrario de la red general	0.0057
ECM arbitrario del entrenamiento	8.3360e-05 = 0.00008336
ECM arbitrario de la validación	0.0452
ECM arbitrario de la prueba	0.0333
ECM de los 18 datos pronosticados	0.0033 (<i>prueba de la hipótesis</i>)

Fuente: Elaboración propia.

De forma gráfica, la estimación A se muestra en la figura 3.5.

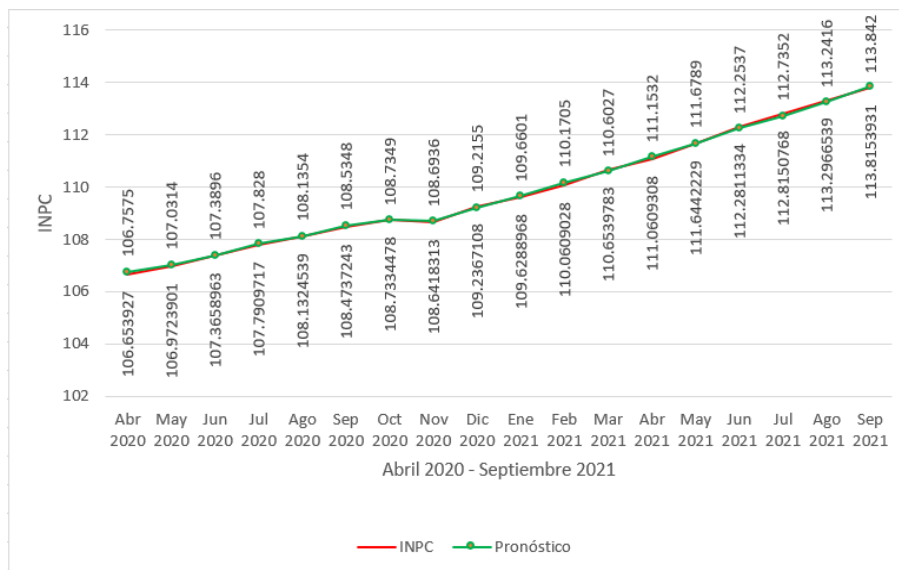


FIGURA 3.5. Pronóstico A del INPC Subyacente, Mediante la Red.
Fuente: Elaboración propia con datos del INEGI y de RNA.

En la figura 3.5 se observa que el pronóstico del INPC, representado por la línea con puntos color verde, se ajusta de forma precisa a la serie original del INPC, denotado con la línea roja.

A continuación, en la tabla 3.10, se muestra el error asociado a cada dato del INPC mensual pronosticado observándose que el error es mínimo en cada evento.

TABLA 3.10. Pronóstico A del INPC.

Fecha	INPC original	Pronóstico del INPC	Error
Abr 2020	106.6539	106.7575	-0.10357
May 2020	106.9724	107.0314	-0.05901
Jun 2020	107.3659	107.3896	-0.0237
Jul 2020	107.791	107.828	-0.03703
Ago 2020	108.1325	108.1354	-0.00295
Sep 2020	108.4737	108.5348	-0.06108
Oct 2020	108.7334	108.7349	-0.00145
Nov 2020	108.6418	108.6936	-0.05177
Dic 2020	109.2367	109.2155	0.021211
Ene 2021	109.6289	109.6601	-0.0312
Feb 2021	110.0609	110.1705	-0.1096
Mar 2021	110.654	110.6027	0.051278
Abr 2021	111.0609	111.1532	-0.09227
May 2021	111.6442	111.6789	-0.03468
Jun 2021	112.2811	112.2537	0.027433
Jul 2021	112.8151	112.7352	0.079877
Ago 2021	113.2967	113.2416	0.055054
Sep 2021	113.8154	113.842	-0.02661

Fuente: Elaboración propia.

Por lo tanto, con los resultados obtenidos por la red para el pronóstico A, respecto al ECM, se comprueba la hipótesis: “Sí, una red neuronal artificial es una herramienta adecuada para pronosticar la inflación subyacente en México, la cual se representa por medio del INPC, entonces será posible de forma empírica, con la información seleccionada para probar los resultados del pronóstico, medir el error del pronóstico y comprobar que se encuentra dentro del intervalo de $\pm 1\%$.”, dado que el ECM que genera la red, 0.0033, se encuentra dentro del intervalo $(-1\%, 1\%)$.

Así, para este primer pronóstico se puede concluir que la red es una herramienta que se puede utilizar para realizar el pronóstico de esta variable económica.

Una vez validada la red óptima, se procede a realizar el pronóstico con TB. Esto es, se aborda la proyección B.

Pronóstico B:

TABLA 3.11. Características Generales de la RNA para Realizar el Pronóstico B.

Periodo de entrenamiento, validación y prueba	enero 1982 – septiembre 2021
Periodo de pronóstico	octubre 2021 – abril 2023
Estructura de la RNA	1 capa oculta con 15 neuronas, 1 capa de salida con 1 neurona; con las funciones de activación tangente sigmoide y lineal, respectivamente
Coef. de aprendizaje	1
Rezagos	70
Límite máximo de la verificación de validación	12
Límite mínimo del gradiente	1.00e-20

Fuente: Elaboración propia.

TABLA 3.12. Error Cuadrático Medio Observado.

ECM arbitrario de la red general	.0056
ECM arbitrario del entrenamiento	.0018
ECM arbitrario de la validación	.0134
ECM arbitrario de la prueba	.0156

Fuente: Elaboración propia.

Con la información de la tabla 3.11, y los mismos parámetros de pesos y bias del pronóstico A, se obtuvo la predicción B a mediano plazo. Los datos se muestran en la tabla 3.13.

TABLA 3.13. Pronóstico B del INPC.

Fecha	Predicción
Oct-21	114.6763
Nov-21	115.4541
Dec-21	116.9893
Jan-22	118.3464
Feb-22	119.4514
Mar-22	120.5663
Apr-22	121.5939
May-22	122.7128
Jun-22	123.9673
Jul-22	124.5527
Aug-22	125.0233
Sep-22	125.5565
Oct-22	126.3307
Nov-22	127.1653
Dec-22	128.4384
Jan-23	129.3611
Feb-23	130.2061
Mar-23	130.8145

Fuente: Elaboración propia.

En la gráfica 3.6, se pueden visualizar los datos del pronóstico B, a medio plazo (octubre 2021-marzo 2023); y a su vez se expone parte de los datos previos del INPC, con el fin de seguir la continuación de la proyección.

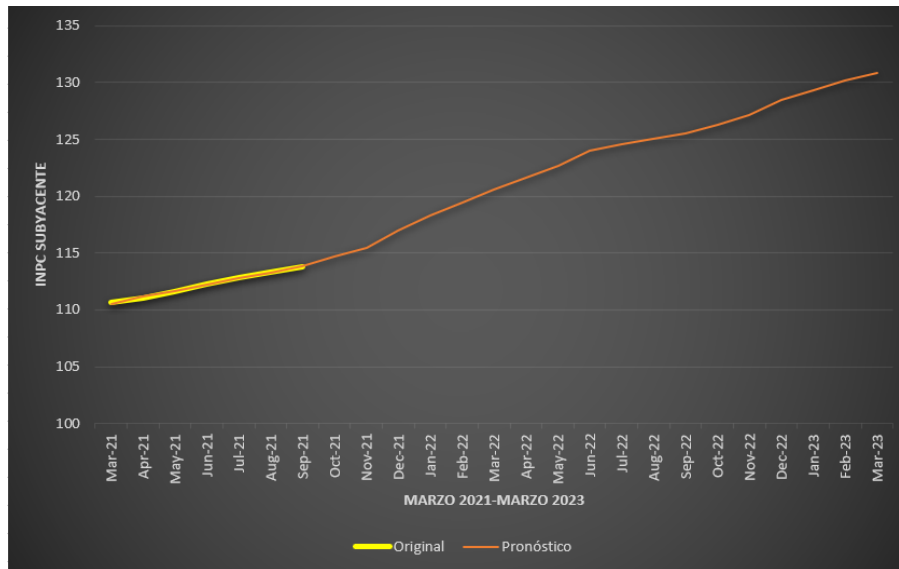


FIGURA 3.6. Pronóstico B del INPC.

Fuente: Elaboración propia con datos del INEGI y de RNA.

De esta forma se concluye el propósito de esta investigación. En el siguiente apartado se trata sobre el cálculo de la tasa de inflación mediante el INPC.

3.5. Tasa de la inflación a través del INPC

La tasa de inflación es la variación porcentual del nivel general de precios [26]. Para calcular dicha tasa de inflación, en términos del INPC, primero se elige un año base, digamos $t - 1$, junto con su $INPC_{t-1}$ respectivo, luego se elige el año con el cual se va a comparar, digamos t , con su respectivo $INPC_t$, así [27, 28]:

$$\pi(t) = \left(\frac{INPC_t - INPC_{t-1}}{INPC_{t-1}} \right) \cdot 100 = \% \text{ de la tasa de inflación respecto al año } t-1 \quad (3.5)$$

A continuación se calcula la tasa de inflación, con los datos obtenidos por la red, correspondientes al pronóstico B. Se utiliza la tasa mensual anualizada. En la tabla 3.14 se muestran los resultados, después de haber aplicado la fórmula:

$$\pi(t) = \left(\frac{INPC_t - INPC_{t-1}}{INPC_{t-1}} \right) \cdot 100 \quad (3.6)$$

donde t denota el año 2022 y $t-1$ el año 2021, con sus respectivos INPC provenientes del pronóstico e INEGI.

TABLA 3.14. Tasa de Inflación Mensual Anualizada Calculada a Partir del INPC pronosticado.

FECHA	Tasa de inflación
2021-Oct	5.4913
2021-Nov	6.2177
2021-Dec	7.1643
2022-Jan	7.9302
2022-Feb	8.4336
2022-Mar	9.0703
2022-Apr	9.3889
2022-May	9.9241
2022-Jun	10.4312
2022-Jul	10.4539
2022-Aug	10.3387
2022-Sep	10.2973
2022-Oct	10.1629
2022-Nov	10.1436
2022-Dec	9.7865
2023-Jan	9.3072

Fuente: Elaboración propia.

Por ejemplo, para el cálculo de la tasa de inflación de enero del 2023, se hace lo siguiente:

$$\pi(2023) = \left(\frac{(129.3611)_{2023} - (118.3464)_{2022}}{(118.3464)_{2022}} \right) \cdot 100 \quad (3.7)$$

TABLA 3.15. Comparación de la tasa de inflación pronosticada vs la original.

Mes	INPC original	INPC pron	Tasa original	Tasa pron
oct-2021	114.3779644	114.6763	5.191150222	5.491279762
nov-2021	114.7990637	115.4541	5.667460049	6.217725264
dic-2021	115.7207054	116.9893	5.93572848	7.164265784
ene-2022	116.4327213	118.3464	6.206232748	7.930167468

Fuente: Elaboración propia con datos del INEGI y de la RNA.

En la tabla 3.15 se muestra una comparación entre la tasa de inflación pronosticada y la “tasa de inflación del INEGI”, mediante la fórmula (3.7).

Como conclusión del presente capítulo, dado que el ECM sobre los 18 datos aproximados del INPC fue de .0033, entonces se concluye que el primer pronóstico es eficiente, debido a que satisface la hipótesis de investigación: el ECM debe de estar en $(-.01, .01)$.

Conclusiones

El entorno económico que prevalece en una economía influye en la toma de decisiones de los diferentes agentes económicos y cuando se habla de decisiones económicas una variable a considerar de gran relevancia es la tasa de inflación. En el presente trabajo de investigación se desarrolló la aplicación de un método de pronóstico de dicha variable aplicando como herramienta una RNA.

El objetivo general que se estableció parte del diseño de una RNA que permitiera pronosticar la inflación⁴ subyacente en México por medio del INPC con un nivel mínimo de error. La hipótesis que guía la presente investigación afirma que, sí, una red neuronal artificial es una herramienta adecuada para pronosticar la inflación subyacente en México, la cual se representa por medio del INPC, entonces sería posible de forma empírica, con la información seleccionada para probar los resultados del pronóstico, medir el error del pronóstico y comprobar que se encuentra dentro del intervalo de $\pm 1\%$.

De acuerdo a lo mencionado respecto al objetivo general y a la hipótesis, en la investigación se da cumplimiento al objetivo general al diseñar una RNA tipo NAR con el algoritmo de entrenamiento de propagación hacia atrás, se demostró que la RNA perceptrón multicapa, permitió simular el comportamiento de la serie de tiempo del INPC subyacente. La hipótesis de investigación se cumplió, dado que el ECM=.0033 de la predicción pertenece al intervalo $\left(-\frac{1}{100}, \frac{1}{100}\right)$.

Los resultados del pronóstico se contrastan con la serie original reflejando valores aproximados; la tendencia del pronóstico es creciente. Obsérvese que el pronóstico B puede verse como una función $f_B : TB \rightarrow INPC$ y hasta continua; de esta forma, se puede considerar el hecho de que $\forall x, y \in TB, f_B(x) < f_B(y)$, i.e. se trataría de una función estrictamente creciente. En sí, preserva la tendencia típica (creciente) del INPC original.

En el contexto actual de la economía mundial la afectación de lo que se conoce como un cisne negro representado por el Síndrome Respiratorio Agudo Severo Coronavirus 2 (SARS-CoV-2) sigue generando incertidumbre en el entorno macroeconómico de los países; en lo que respecta a la inflación en México se obtienen desafíos importantes al presentarse presiones inflacionarias con un crecimiento estancado y tasas de desempleo elevadas, fenómeno que se conoce como estanflación [63] aunado ahora con las

⁴En sí, la inflación que se utilizó fue la inflación, particular, que se obtuvo a partir de la fórmula (3.5), y tomando los valores del INPC subyacentes pronosticados (pronóstico B).

cuestiones geopolíticas que prevalecen y que tienen impacto en la subida del petróleo y sus derivados y algunos energéticos; el resultado del pronóstico no está tan alejado del contexto económico.

El ejercicio que se llevó a cabo en este trabajo de investigación permite reafirmar a una RNA como una herramienta que se puede utilizar para realizar pronósticos. Los resultados no difieren del encontrado por otros investigadores como es el caso de [49] en el que se pronosticó el precio de condominios utilizando RNA (perceptrón multicapa) y técnicas estadísticas tradicionales, concluyendo que de acuerdo a la RECM, el modelo más eficiente fue el perceptrón multicapa. Por otro lado en [50] se comparan cuatro métodos distintos: SVM, ARM, ARIMA y RNA (perceptrón multicapa), para el pronóstico de la demanda de turismo en Indonesia y se encontró que el ECM y el porcentaje de precisión óptimo se obtuvo con el modelo del perceptrón multicapa.

La aplicación de técnicas matemáticas al pronóstico de una variable económica adquiere relevancia al existir la necesidad de tener información oportuna que guíe la toma de decisiones, ya que, como agentes racionales, la toma de decisiones no es en base a improvisaciones sino en predicciones sustentadas, para saber cómo planificar y qué acciones tomar al respecto desde hoy con el fin de afrontar las situaciones futuras de forma oportuna. La técnica expuesta en esta investigación representa una alternativa viable de aplicación al pronóstico de cualquier variable macroeconómica que puede ser útil para su estudio contribuyendo a la generación de información pertinente a la toma de decisiones financieras y económicas.

La incorporación de herramientas modernas como las redes neuronales artificiales, que a pesar de existir desde hace varias décadas no se incorporan en los planes de estudio tradicionales y en los cursos de pronóstico, es un reto interesante para la superación académica y una oportunidad para investigar, aplicar y evaluar las técnicas de pronóstico en un ambiente real.

En investigaciones futuras, existe un campo de oportunidad al incluir otro tipo de redes neuronales para elaborar pronósticos, así como también otro tipo de variables, inclusive de otras áreas: médicas, físicas, astronómicas, etc. Y en general continuar aplicando herramientas de inteligencia artificial dentro de los modelos económicos matemáticos que parten de la abstracción de un problema real y se fundamentan en una teoría económica reforzándose con el desarrollo matemático, para así dar solución al problema planteado.

Apéndice

Apéndice A: Datos del INPC

FECHA	INPC
Ene 1982	0.092901
Feb 1982	0.096869
Mar 1982	0.101014
Abr 1982	0.106836
May 1982	0.113471
Jun 1982	0.119429
Jul 1982	0.125428
Ago 1982	0.137414
Sep 1982	0.144974
Oct 1982	0.15343
Nov 1982	0.162443
Dic 1982	0.174668
Ene 1983	0.195202
Feb 1983	0.207338
Mar 1983	0.217868
Abr 1983	0.230276
May 1983	0.240226
Jun 1983	0.249485
Jul 1983	0.26413
Ago 1983	0.276613
Sep 1983	0.285926
Oct 1983	0.295174
Nov 1983	0.312317
Dic 1983	0.325912
Ene 1984	0.34335
Feb 1984	0.363431
Mar 1984	0.378914
Abr 1984	0.391804
May 1984	0.403228
Jun 1984	0.419733
Jul 1984	0.434772
Ago 1984	0.447045
Sep 1984	0.463326
Oct 1984	0.482508
Nov 1984	0.498635
Dic 1984	0.519677
Ene 1985	0.546939
Feb 1985	0.571629
Mar 1985	0.597757
Abr 1985	0.614614
May 1985	0.632167
Jun 1985	0.650252
Jul 1985	0.673014
Ago 1985	0.703559
Sep 1985	0.734981
Oct 1985	0.766076
Nov 1985	0.802321
Dic 1985	0.83813
Ene 1986	0.909993
Feb 1986	0.957545
Mar 1986	1.008175
Abr 1986	1.063468
May 1986	1.125497
Jun 1986	1.20465
Jul 1986	1.265074
Ago 1986	1.337383
Sep 1986	1.421436
Oct 1986	1.515087
Nov 1986	1.617609
Dic 1986	1.742314
Ene 1987	1.886889
Feb 1987	2.029259
Mar 1987	2.172018
Abr 1987	2.329889
May 1987	2.498393

Jun 1987	2.684685
Jul 1987	2.915519
Ago 1987	3.129306
Sep 1987	3.362996
Oct 1987	3.657629
Nov 1987	3.997515
Dic 1987	4.495541
Ene 1988	5.089183
Feb 1988	5.618016
Mar 1988	5.953512
Abr 1988	6.125769
May 1988	6.244011
Jun 1988	6.329859
Jul 1988	6.391669
Ago 1988	6.455503
Sep 1988	6.470215
Oct 1988	6.511381
Nov 1988	6.56647
Dic 1988	6.687693
Ene 1989	6.90067
Feb 1989	7.039663
Mar 1989	7.13621
Abr 1989	7.222918
May 1989	7.29814
Jun 1989	7.366201
Jul 1989	7.43351
Ago 1989	7.520436
Sep 1989	7.635181
Oct 1989	7.77111
Nov 1989	7.891836
Dic 1989	8.121922
Ene 1990	8.435019
Feb 1990	8.590027
Mar 1990	8.721044
Abr 1990	8.892265
May 1990	9.097511

Jun 1990	9.325962
Jul 1990	9.512833
Ago 1990	9.684587
Sep 1990	9.842794
Oct 1990	10.0009
Nov 1990	10.21673
Dic 1990	10.50748
Ene 1991	10.78754
Feb 1991	10.99653
Mar 1991	11.16022
Abr 1991	11.28235
May 1991	11.38954
Jun 1991	11.49724
Jul 1991	11.59955
Ago 1991	11.70555
Sep 1991	11.86846
Oct 1991	12.0324
Nov 1991	12.14651
Dic 1991	12.3622
Ene 1992	12.59039
Feb 1992	12.72887
Mar 1992	12.84409
Abr 1992	12.96353
May 1992	13.12111
Jun 1992	13.25159
Jul 1992	13.35994
Ago 1992	13.4764
Sep 1992	13.62694
Oct 1992	13.71768
Nov 1992	13.81569
Dic 1992	13.9521
Ene 1993	14.12747
Feb 1993	14.28536
Mar 1993	14.43021
Abr 1993	14.5263
May 1993	14.59082

Jun 1993	14.65334
Jul 1993	14.71408
Ago 1993	14.80893
Sep 1993	14.92661
Oct 1993	15.01512
Nov 1993	15.06983
Dic 1993	15.12623
Ene 1994	15.22566
Feb 1994	15.35368
Mar 1994	15.46655
Abr 1994	15.56369
May 1994	15.65291
Jun 1994	15.73489
Jul 1994	15.79002
Ago 1994	15.86928
Sep 1994	15.99805
Oct 1994	16.07944
Nov 1994	16.16266
Dic 1994	16.28816
Ene 1995	16.89033
Feb 1995	17.66232
Mar 1995	18.49777
Abr 1995	19.89987
May 1995	20.88286
Jun 1995	21.59524
Jul 1995	22.04632
Ago 1995	22.45676
Sep 1995	22.98212
Oct 1995	23.38381
Nov 1995	23.8604
Dic 1995	24.53024
Ene 1996	25.32584
Feb 1996	25.92611
Mar 1996	26.53114
Abr 1996	27.26187
May 1996	27.75748

Jun 1996	28.23469
Jul 1996	28.65892
Ago 1996	29.04486
Sep 1996	29.55244
Oct 1996	29.88771
Nov 1996	30.24775
Dic 1996	30.89511
Ene 1997	31.66658
Feb 1997	32.21644
Mar 1997	32.62906
Abr 1997	33.04308
May 1997	33.46857
Jun 1997	33.82355
Jul 1997	34.10109
Ago 1997	34.37463
Sep 1997	34.86683
Oct 1997	35.18775
Nov 1997	35.53675
Dic 1997	35.9815
Ene 1998	36.66825
Feb 1998	37.3774
Mar 1998	37.79304
Abr 1998	38.225
May 1998	38.63612
Jun 1998	39.09242
Jul 1998	39.39367
Ago 1998	39.79928
Sep 1998	40.56582
Oct 1998	41.13432
Nov 1998	41.60479
Dic 1998	42.25498
Ene 1999	43.27158
Feb 1999	44.06277
Mar 1999	44.75777
Abr 1999	45.29382
May 1999	45.73296

Jun 1999	46.08845
Jul 1999	46.39267
Ago 1999	46.74352
Sep 1999	47.31303
Oct 1999	47.65321
Nov 1999	47.95593
Dic 1999	48.34313
Ene 2000	48.96917
Feb 2000	49.50797
Mar 2000	49.82817
Abr 2000	50.08998
May 2000	50.3091
Jun 2000	50.47798
Jul 2000	50.64577
Ago 2000	50.85747
Sep 2000	51.24921
Oct 2000	51.51716
Nov 2000	51.84465
Dic 2000	52.13669
Ene 2001	52.52422
Feb 2001	52.92035
Mar 2001	53.23714
Abr 2001	53.4936
May 2001	53.68653
Jun 2001	53.85885
Jul 2001	53.94192
Ago 2001	54.12178
Sep 2001	54.53141
Oct 2001	54.66885
Nov 2001	54.79951
Dic 2001	54.93076
Ene 2002	55.20477
Feb 2002	55.59532
Mar 2002	55.82166
Abr 2002	56.02071
May 2002	56.12432

Jun 2002	56.19529
Jul 2002	56.27083
Ago 2002	56.46041
Sep 2002	56.83222
Oct 2002	56.96475
Nov 2002	57.07456
Dic 2002	57.19333
Ene 2003	57.44781
Feb 2003	57.73289
Mar 2003	57.96746
Abr 2003	58.16381
May 2003	58.3181
Jun 2003	58.38159
Jul 2003	58.46081
Ago 2003	58.67014
Sep 2003	58.95136
Oct 2003	59.07789
Nov 2003	59.2331
Dic 2003	59.40446
Ene 2004	59.6331
Feb 2004	59.90185
Mar 2004	60.12152
Abr 2004	60.31955
May 2004	60.45874
Jun 2004	60.61955
Jul 2004	60.72514
Ago 2004	60.8939
Sep 2004	61.24245
Oct 2004	61.41835
Nov 2004	61.56421
Dic 2004	61.73593
Ene 2005	61.96721
Feb 2005	62.2078
Mar 2005	62.38519
Abr 2005	62.50379
May 2005	62.62479

Jun 2005	62.77681
Jul 2005	62.90595
Ago 2005	62.99278
Sep 2005	63.29559
Oct 2005	63.44667
Nov 2005	63.57726
Dic 2005	63.75689
Ene 2006	63.89572
Feb 2006	64.10456
Mar 2006	64.35873
Abr 2006	64.57716
May 2006	64.66446
Jun 2006	64.85981
Jul 2006	65.02752
Ago 2006	65.14082
Sep 2006	65.49716
Oct 2006	65.64849
Nov 2006	65.82049
Dic 2006	66.08097
Ene 2007	66.38775
Feb 2007	66.64028
Mar 2007	66.83666
Abr 2007	66.96598
May 2007	67.09843
Jun 2007	67.29018
Jul 2007	67.49456
Ago 2007	67.6772
Sep 2007	68.00091
Oct 2007	68.22375
Nov 2007	68.37848
Dic 2007	68.63982
Ene 2008	68.93751
Feb 2008	69.23388
Mar 2008	69.57148
Abr 2008	69.84505
May 2008	70.17806

Jun 2008	70.50768
Jul 2008	70.79065
Ago 2008	71.08304
Sep 2008	71.49797
Oct 2008	71.70265
Nov 2008	72.02911
Dic 2008	72.44065
Ene 2009	72.75882
Feb 2009	73.09727
Mar 2009	73.47385
Abr 2009	73.7848
May 2009	73.93156
Jun 2009	74.11789
Jul 2009	74.34485
Ago 2009	74.45921
Sep 2009	74.7536
Oct 2009	74.99505
Nov 2009	75.1029
Dic 2009	75.45225
Ene 2010	76.01922
Feb 2010	76.33328
Mar 2010	76.60114
Abr 2010	76.68369
May 2010	76.8625
Jun 2010	76.96833
Jul 2010	77.12871
Ago 2010	77.2034
Sep 2010	77.48281
Oct 2010	77.63094
Nov 2010	77.81495
Dic 2010	78.15082
Ene 2011	78.50811
Feb 2011	78.82409
Mar 2011	79.05936
Abr 2011	79.11899
May 2011	79.25986

Jun 2011	79.41323	Jun 2014	87.09625
Jul 2011	79.59164	Jul 2014	87.25995
Ago 2011	79.68702	Ago 2014	87.44522
Sep 2011	79.89896	Sep 2014	87.70142
Oct 2011	80.10607	Oct 2014	87.84749
Nov 2011	80.36458	Nov 2014	87.98924
Dic 2011	80.77127	Dic 2014	88.1935
Ene 2012	81.13318	Ene 2015	88.16443
Feb 2012	81.47818	Feb 2015	88.46418
Mar 2012	81.67729	Mar 2015	88.69616
Abr 2012	81.80032	Abr 2015	88.83426
May 2012	82.01758	May 2015	88.93711
Jun 2012	82.19464	Jun 2015	89.12294
Jul 2012	82.45063	Jul 2015	89.27674
Ago 2012	82.63563	Ago 2015	89.456
Sep 2012	82.78193	Sep 2015	89.78776
Oct 2012	82.97047	Oct 2015	90.01353
Nov 2012	83.01338	Nov 2015	90.04518
Dic 2012	83.11374	Dic 2015	90.32069
Ene 2013	83.46649	Ene 2016	90.49499
Feb 2013	83.89126	Feb 2016	90.81943
Mar 2013	84.14365	Mar 2016	91.14475
Abr 2013	84.211	Abr 2016	91.34575
May 2013	84.37726	May 2016	91.541
Jun 2013	84.48557	Jun 2016	91.77385
Jul 2013	84.51427	Jul 2016	91.93052
Ago 2013	84.59064	Ago 2016	92.10472
Sep 2013	84.86465	Sep 2016	92.5434
Oct 2013	85.02768	Oct 2016	92.80215
Nov 2013	85.14266	Nov 2016	93.01043
Dic 2013	85.42172	Dic 2016	93.42534
Ene 2014	86.1499	Ene 2017	93.96614
Feb 2014	86.39256	Feb 2017	94.68426
Mar 2014	86.57812	Mar 2017	95.22652
Abr 2014	86.83009	Abr 2017	95.65483
May 2014	86.91155	May 2017	95.9181

Jun 2017	96.21005
Jul 2017	96.46777
Ago 2017	96.71268
Sep 2017	96.98562
Oct 2017	97.23048
Nov 2017	97.56524
Dic 2017	97.97401
Ene 2018	98.25201
Feb 2018	98.73169
Mar 2018	99.05736
Abr 2018	99.20321
May 2018	99.45811
Jun 2018	99.68833
Jul 2018	99.97429
Ago 2018	100.2203
Sep 2018	100.5448
Oct 2018	100.8566
Nov 2018	101.1104
Dic 2018	101.5825
Ene 2019	101.7848
Feb 2019	102.2242
Mar 2019	102.5761
Abr 2019	103.0433
May 2019	103.2122
Jun 2019	103.5256
Jul 2019	103.7944
Ago 2019	104.0054
Sep 2019	104.3153
Oct 2019	104.572
Nov 2019	104.8037
Dic 2019	105.2338
Ene 2020	105.5771
Feb 2020	105.9614
Mar 2020	106.2725
Abr 2020	106.6539
May 2020	106.9724

Jun 2020	107.3659
Jul 2020	107.791
Ago 2020	108.1325
Sep 2020	108.4737
Oct 2020	108.7334
Nov 2020	108.6418
Dic 2020	109.2367
Ene 2021	109.6289
Feb 2021	110.0609
Mar 2021	110.654
Abr 2021	111.0609
May 2021	111.6442
Jun 2021	112.2811
Jul 2021	112.8151
Ago 2021	113.2967
Sep 2021	113.8154

Elaboración propia con datos del INEGI.

Apéndice B: Código NAR (original) proporcionado por MATLAB, generado por la aplicación Neural Net Time Series

Partes del siguiente código, fueron con las que se partió para elaborar los pronósticos del INPC. Y después de múltiples modificaciones y pruebas se obtuvo el código C.

.....Código generado mediante ntstool.....

```
% Solve an Autoregression Time-Series Problem with a NAR Neural Network
% Script generated by Neural Time Series app
% Created 04-Feb-2015 12:03:04
%
% This script assumes this variable is defined:
%
% T - feedback time series.

T = T;

% Choose a Training Function
% For a list of all training functions type: help ntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Nonlinear Autoregressive Network
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narnet(feedbackDelays,hiddenLayerSize,'open',trainFcn);

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer
% states. Using PREPARETS allows you to keep your original time series data
% unchanged, while easily customizing it for networks with differing
% numbers of delays, with open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,{}, {},T);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
```

```

%figure, plotresponse(t,y)
%figure, ploterrcorr(e)
%figure, plotinerrcorr(x,e)

% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the output layer.
netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
view(netc)
[xc,xic,aic,tc] = preparets(netc,{}, {},T);
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(net,tc,yc)

% Step-Ahead Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is
% given y(t+1). For some applications such as decision making, it would
% help to have predicted y(t+1) once y(t) is available, but before the
% actual y(t+1) occurs. The network can be made to return its output a
% timestep early by removing one delay so that its minimal tap delay is now
% 0 instead of 1. The new network returns the same outputs as the original
% network, but outputs are shifted left one timestep.
nets = removedelay(net);
nets.name = [net.name ' - Predict One Step Ahead'];
view(nets)
[xs,xis,ais,ts] = preparets(nets,{}, {},T);
ys = nets(xs,xis,ais);
stepAheadPerformance = perform(nets,ts,ys)

```

Apéndice C: código C para el pronóstico A y B i.e. código completo

```

%OCTOBER_inpc es el nombre del script en MATLAB.

%pronostico para comprobar la hipotesis:
t=xlsread('/Users/VeronicaRM/INPCoct_28.xlsx','Sheet1','d4:d462');
%t deja los ultimos 18 datos, porque con esos se medira el ECM.
%t contiene todos los datos a considerar para generar el pronostico
%de 18 datos
T = tonndata(t,false,false);

tt=xlsread('/Users/VeronicaRM/INPCoct_28.xlsx','Sheet1','d463:d480');
%este elemento de arriba contiene los ultimos 18 datos
TT = tonndata(tt,false,false);

%pronostico para los 18 datos futuros: de oct. 2021 a marzo 2023.
todos=xlsread('/Users/VeronicaRM/INPCoct_28.xlsx','Sheet1','d4:d480');
%se consideran todos los datos del INPC
TODOS = tonndata(todos,false,false);

trainFcn = 'trainlm';

num=70;
feedbackDelays = 1:num;

```

```

hiddenLayerSize = 15;

net = narnet(feedbackDelays,hiddenLayerSize,'open',trainFcn);

[x,xi,ai,tn] = preparets(net,{}, {},T);
[xT,xiT,aiT,tnT] = preparets(net,{}, {},TODOS);

net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

net.trainParam.lr=1;
%net.divideMode='value';

%Valor minimo del gradiente
net.trainParam.min_grad=1e-20;

%For validation checks (maximo numero):
net.trainParam.max_fail=12;

%.6 es mejor a .1, definitivamente
net.trainParam.epochs=50000;

[net,tr] = train(net,x,tn,xi,ai);
[net,trB] = train(net,xT,tnT,xiT,aiT);

%.....
%...PRONOSTICO.....
%A
nets = removedelay(net);
[xs,xis,ais,ts] = preparets(nets,{}, {},T);
pro1=nets(xs,xis,ais);
TamPro1ConUnPrimerPronostico=size(pro1);

V=T;

xa=size(t);
xb=xa-num;

for i=1:17
    V(xa(1)+i)=pro1(xb(1)+i);
    [xs2,xis2,ais2,ts2] = preparets(nets,{}, {},V);
    pro1=nets(xs2,xis2,ais2);
end
b=cell2mat(pro1(end-17:end));
bb=reshape(b,18,1);
display(bb,'18 datos pronosticados')

%.....
%B
nets = removedelay(net);
[xsT,xisT,aisT,tsT] = preparets(nets,{}, {},TODOS);
proB=nets(xsT,xisT,aisT);

```

```

VB=TODO;

xa=size(todos);
xb=xa-num;

for i=1:17
    VB(xa(1)+i)=proB(xb(1)+i);
    [xs2T,xis2T,ais2T,ts2T] = preparets(nets,{}, {},VB);
    proB=nets(xs2T,xis2T,ais2T);
end
bB=cell2mat(proB(end-17:end));
bbB=reshape(bB,18,1);
display(bbB,'18 datos pronosticados: pro B')

%.....END PRONOSTICO.....
%.....

    e18=gsubtract(TT,pro1(end-17:end));
    ECM=mse(e18);
    display(ECM,'ECM de los 18 datos')

%.....
%...WEIGHTS & BIAS.....
W1=net.IW{1};
display(W1,'Pesos: de los inputs a la primer capa')

Wb=net.LW{2};
display(Wb,'Pesos: de la primer capa a la capa de salida')

b1=net.b{1};
display(b1,'Bias de la primer capa oculta')
b2=net.b{2};
display(b2,'Bias de la capa de salida')

%.....
%...END WEIGHTS & BIAS.....

%.....
%.....LOS WHILES.....

%Para la generación de datos de la tabla 3.3:
%Poner iter=1; minimo=20000; OCTOBER_inpc (en command windows)

% while(ECM>.01 && iter<20)
%     iter=iter+1;
%     if ECM<minimo
%         minimo=ECM;
%     end
%     OCTOBER_inpc;
% end

% if(ECM <minimo)
%     minimo=ECM;
%     display(minimo)

```



```

% else
%   display(minimo)
% end

%Para conseguir cumplir la hipótesis, dentro de un límite
%de 5,000 ejecuciones, y contar el núm.
%de iteraciones que necesito:
%Poner iter=1; minimo=1; set(0,'RecursionLimit',5000); OCTOBER_inpc
%(en command windows)

while(ECM>.01)
    iter=iter+1;
    if ECM<minimo
        minimo=ECM;
    end
    OCTOBER_inpc;
end
minimo=ECM;
display(minimo)
%.....END LOS WHILES.....
%.....

```

Referencias

- [1] INEGI, *Índice nacional de precios al consumidor primera quincena de enero de 2022*, (ene. de 2022), URL: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2022/inpc_1q/inpc_1q2022_01.pdf.
- [2] Puma Capitalista, *Cómo curar la inflación | Milton Friedman – Libre para elegir*, (2011), URL: <https://www.youtube.com/watch?v=ArBdqddGGcM>.
- [3] Carlos Torres Rosas, *Primer informe mensual de resultados de la economía nacional. Conferencia presidente AMLO*, Discurso por el presidente Andrés Manuel López Obrador, (2022), URL: <https://www.youtube.com/watch?v=yX-rblGJFQ>.
- [4] Howard B Demuth y col., *Neural network design*, Martin Hagan, (2014).
- [5] Lilia Alejandra Flores Castillo, «Pronóstico del Índice Nacional de Precios al Consumidor/Forecast of the National Index of Consumer Prices», en: *Revista Iberoamericana de Contaduría, Economía y Administración: RICEA* 6.12 (2017) págs. 60-88.
- [6] Wikipedia, *Alan Turing*, (2021), URL: https://es.wikipedia.org/wiki/Alan_Turing.
- [7] Damián Jorge Matich, «Redes Neuronales: Conceptos básicos y aplicaciones», en: *Universidad Tecnológica Nacional, México* (2001).
- [8] Eduardo Rivera y col., «Introducción a las Redes Neuronales Artificiales.», en: (2015).
- [9] Wikipedia, *Donald Hebb*, (2020), URL: https://es.wikipedia.org/wiki/Donald_Hebb.
- [10] Xavier Basogain, «Redes neuronales artificiales y sus aplicaciones», en: *Dpto. Ingeniería de Sistemas y Automática, Escuela Superior de Ingeniería Bilbao. Open Course Ware.[En línea] disponible en http://ocw.ehu.es/enseñanzas-tecnicas/redes-neuronales-artificiales-y-sus-aplicaciones/Course_listing.[Consultada 20-09-2012]* (2008).
- [11] Glosser.ca, *Colored Neural Network*, (2013), URL: https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Colored_neural_network.svg.
- [12] Daniel Shiffman, «The Nature of Code: Chapter 10», en: *Neural Networks* (2012).

- [13] Jorge de Andrés Sánchez, «Dos aplicaciones empíricas de las redes neuronales artificiales a la clasificación y la predicción financiera en el mercado español», en: *RAE: Revista Asturiana de Economía* 28 (2003) págs. 61-87.
- [14] Javier Del Carpio Gallegos, «Las redes neuronales artificiales en las finanzas», en: (2011).
- [15] Ernesto Varela-Arregoces y Edwin Campbells, «Redes Neuronales Artificiales: una revisión del estado del arte, aplicaciones y tendencias futuras», en: *Investigación y desarrollo en TIC* 2.1 (2011) págs. 18-27.
- [16] deeplizard, *Bias in an Artificial Neural Network explained | How bias impacts training*, YouTube, (abr. de 2018), URL: <https://www.youtube.com/watch?v=HetFihSXSys>.
- [17] Daniel Shiffman, *10.12: Neural Networks: Feedforward Algorithm Part 1 - The Nature of Code*, The Coding Train channel from YouTube, (2018), URL: <https://www.youtube.com/watch?v=qWK7yW8oS0I>.
- [18] Xpikuos, *Perceptrón redes neuronales : capas ocultas, hiperparámetros , neuronas por capa (2021) | IA*, (2018), URL: <https://www.youtube.com/watch?v=HKPE3mrxOPo>.
- [19] Man-Wai Mak, WG Allen y GG Sexton, «Speaker identification using multilayer perceptrons and radial basis function networks», en: *Neuro-computing* 6.1 (1994) págs. 99-117.
- [20] Wikipedia, *Multilayer perceptron*, (2021), URL: https://en.wikipedia.org/wiki/Multilayer_perceptron.
- [21] George Cybenko, «Approximation by superpositions of a sigmoidal function», en: *Mathematics of control, signals and systems* 2.4 (1989) págs. 303-314.
- [22] Oscar Alsing Ferreira, *But How Does The MultiLayer Perceptron Actually Work?*, (2017), URL: <https://www.youtube.com/watch?v=AZEfmoWBXwg>.
- [23] David E Rumelhart, Geoffrey E Hinton y Ronald J Williams, *Learning internal representations by error propagation*, inf. téc., California Univ San Diego La Jolla Inst for Cognitive Science, (1985).
- [24] Dot CSV, *¿Qué es una Red Neuronal? Parte 3 : Backpropagation | DotCSV*, (2018), URL: https://www.youtube.com/watch?v=eNlqz_noix8.
- [25] INEGI, *Preguntas frecuentes*, (2018), URL: <https://www.inegi.org.mx/programas/inpc/2018/PreguntasF/>.
- [26] N Gregory Mankiw, *Macroeconomía*, Antoni Bosch editor, (2014).
- [27] Economía y desarrollo, *What is the Consumer Price Index (CPI) and how is it measured? | Chapter 3 - Macroeconomics*, (2016), URL: <https://www.youtube.com/watch?v=GPIWpvhKjo8&t=201s>.
- [28] GREGORY Mankiw, «Principles of economics», en: *Mason: Cengage Learning* (2011).
- [29] Olivier Blanchard, Amighini Alessia y Giavazzi Francesco, *Macroeconomía*, Pearson Educación, (2012).
- [30] Banco de México, *Sistema de información económica*, (2011), URL: <https://www.banxico.org.mx/tipcamb/main.do?page=inf&idioma=sp>.

- [31] INEGI, *Calculadora de inflación*, (2011), URL: <https://www.inegi.org.mx/app/indicesdeprecios/calculadorainflacion.aspx>.
- [32] BANXICO-educ, *¿Sabías que hay precios más sensibles a cambios que otros?*, Inflación, Español, URL: http://educ.banxico.org.mx/infografias_y_fichas/inflacion_infografias_/sabias-que-hay-precios-mas-se.html.
- [33] Wikipedia, *Joseph Lowe (economist)*, (2021), URL: [https://en.wikipedia.org/wiki/Joseph_Lowe_\(economist\)](https://en.wikipedia.org/wiki/Joseph_Lowe_(economist)).
- [34] OIT / FMI / OCDE / Oficina Estadística de las Comunidades Europeas / Organización de las Naciones Unidas / Banco Mundial, *Manual del índice de precios al consumidor: Teoría y práctica*, Departamento de Tecnología y Servicios Generales Washington, Fondo Monetario Internacional, 2006, (2006).
- [35] Wikipedia, *Formal calculation*, (2021), URL: https://en.wikipedia.org/wiki/Price_index#Formal_calculation.
- [36] Economía y Desarrollo, *What is the Consumer Price Index (CPI) and how is it measured? | Chapter 3 - Macroeconomics*, (2016), URL: <https://www.youtube.com/watch?v=GPIWpvhKjo8>.
- [37] Wikipedia, *Laspeyres*, (2021), URL: https://en.wikipedia.org/wiki/%5C%C3%5C%89tienne_Laspeyres.
- [38] INEGI, *Índice Nacional de Precios al Consumidor: documento metodológico: base segunda quincena de julio de 2018*, (2018).
- [39] Wikipedia, *Relative ease of calculating the Laspeyres index*, (2021), URL: https://en.wikipedia.org/wiki/Price_index#Relative_ease_of_calculating_the_Laspeyres_index.
- [40] WE Diewert, «Chapter 5:Índex Numbers», en: *Essays in Index Number Theory*. eds WE Diewert and AO Nakamura 1 (1993).
- [41] tdm, *About the 9 tests on Index number theory to evaluate the efficiency of the different types of price indexes*, (2021), URL: <https://economics.stackexchange.com/a/45445/>.
- [42] International Monetary Fund, *Revised PPI Manual, Elementary indices*, (ago. de 2004), URL: <https://www.imf.org/external/np/sta/teggppi/ch20.pdf>.
- [43] Julián Pérez Porto y Ana Gardey, *Definición de estrato*, (2010), URL: <https://definicion.de/estrato/>.
- [44] INEGI, *Lineamientos para el ciclo de actualización de la información económica generada por la dirección general de estadísticas económicas del INEGI*, (2015).
- [45] Héctor Guillén Romo, *Orígenes de la crisis en México: inflación y endeudamiento externo [1940-1982]*, vol. 34, Ediciones era, (1984).
- [46] Carlos Tello, «Estancamiento económico, desigualdad y pobreza: 1982-2009», en: *Economía UNAM* 7.19 (2010) págs. 5-44.
- [47] Anthony Daniel Mora Saavedra, «Utilización de Redes Neuronales Artificiales (RNA) como alternativa para el pronóstico del Índice de Precios al Consumidor (IPC) y contraste de resultados con modelos estadísticos tradicionales (ARIMA y VAR) para el Ecuador periodo 2000-2017.», en: (2018).

- [48] Pablo González Peláez y col., «Pronóstico del IPC mediante una red neuronal artificial borrosa», en: (2019).
- [49] Wan Teng Lim y col., «Housing price prediction using neural networks», en: *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, IEEE, (2016), págs. 518-522.
- [50] Andry Alamsyah y Putu Bella Ayastri Friscintia, «Artificial neural network for Indonesian tourism demand forecasting», en: *2019 7th International Conference on Information and Communication Technology (ICoICT)*, IEEE, (2019), págs. 1-7.
- [51] INEGI, *Índice Nacional de Precios al Consumidor (INPC)*, (2021), URL: <https://www.inegi.org.mx/temas/inpc/>.
- [52] MATLAB: Neural Time Series (ntstool), *Para encontrar a la red NAR: MATLAB → APPS → MATH, STATISTICS AND OPTIMIZATION → Neural Net Time Series → NAR*. English, (2015).
- [53] Bobyandbob, *How to add bias and weight to neural network diagram?*, Stack Exchange Inc., (nov. de 2017), URL: <https://tex.stackexchange.com/a/401865>.
- [54] Ron Barry, *When would one use a 'tanh' transfer function in the hidden layer of a neural network?*, English, Quora, (2017), URL: <https://www.quora.com/When-would-one-use-a-tanh-transfer-function-in-the-hidden-layer-of-a-neural-network>.
- [55] Rukshan Pramoditha, *How to Choose the Right Activation Function for Neural Networks*, English, Towards Data Science, (ene. de 2022), URL: <https://towardsdatascience.com/how-to-choose-the-right-activation-function-for-neural-networks-3941ff0e6f9c>.
- [56] MathWorks, *init*, (2021), URL: <https://www.mathworks.com/help/deeplearning/ref/init.html>.
- [57] MathWorks, *initnw*, (2021), URL: <https://www.mathworks.com/help/deeplearning/ref/initnw.html;jsessionid=83cc68494db1685afa08be555194>.
- [58] Wikipedia, *Sigmoid function->Examples->Hyperbolic tangent*, (2021), URL: https://en.wikipedia.org/wiki/Sigmoid_function.
- [59] MathWorks, *tansig*, Help Center, English, The MathWorks, Inc., (2021), URL: <https://www.mathworks.com/help/deeplearning/ref/tansig.html>.
- [60] Camilo Dagum, «Distribución del rezago y dinámica económica», en: *Revista de Economía y Estadística* 11.1-2 (1967) págs. 101-110.
- [61] Jacob Steeves, *NARX input delays what exactly do they mean?*, MATLAB Answers, English, The MathWorks, Inc., (2014), URL: <https://www.mathworks.com/matlabcentral/answers/164784-narx-input-delays-what-exactly-do-they-mean>.
- [62] Greg Heath, *NARX input delays what exactly do they mean?*, English, The MathWorks, Inc., (dic. de 2014), URL: <https://www.mathworks.com/matlabcentral/answers/164784-narx-input-delays-what-exactly-do-they-mean>.

- [63] Alejandro Padilla, *El fantasma de la estanflación ronda de nuevo*, (2022), URL: <https://www.forbes.com.mx/el-fantasma-de-la-estanflacion-ronda-de-nuevo/>.