

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN - TACNA

Facultad de Ingeniería

Escuela Profesional de Ingeniería en Informática y Sistemas

USO DE REDES NEURONALES ARTIFICIALES PARA EL  
PRONÓSTICO DE DEMANDA DE HIELO INDUSTRIAL  
EN HIELOS SANTA LUCÍA S.R.L. - 2015

**TESIS**

Presentada por:

**Bach. Andrea Isabel Montes Albarracin**

Para optar el Título Profesional de:

**INGENIERO EN INFORMÁTICA Y SISTEMAS**

TACNA - PERÚ

2016

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN – TACNA

FACULTAD DE INGENIERÍA

JURADO CALIFICADOR Y CALIFICACIÓN DE LA SUSTENTACIÓN DE TESIS

TESIS N° \_\_\_\_\_

TITULO PROFESIONAL DE:  
Ingeniero en Informática y Sistemas

La Secretaría Académico de la Facultad de Ingeniería, por resolución de Facultad N°03256-2015-FAIN/UNJBG, designó Jurado para la sustentación oral de la Tesis titulada: “USO DE REDES NEURONALES ARTIFICIALES PARA EL PRONÓSTICO DE DEMANDA DE HIELO INDUSTRIAL EN HIELOS SANTA LUCÍA S.R.L. - 2015”.

El mismo que está conformado por:

Presidente : Mgr. Erbert Francisco Osco Mamani  
Secretario : Ing. Gianfranco Alexey Málaga Tejada  
Vocal : MSc. Edgar Aurelio Taya Acosta

Para calificar la sustentación de la Tesis en acto público el día 18 de diciembre del 2015.

Presentado por la Bachiller Andrea Isabel Montes Albarracin, de la Escuela Profesional de Ingeniería en Informática y Sistemas.

El Jurado Calificador en forma secreta e individual emitió su opinión sobre el tema de la tesis expuesta y procedió a obtener el promedio que arrojó el calificativo de aprobado con la nota de Dieciocho (18) – Promedio Sobresaliente.

Para ratificar lo detallado firman:



Mgr. Erbert Francisco Osco Mamani  
Presidente



Ing. Gianfranco Alexey Málaga Tejada  
Secretario



MSc. Edgar Aurelio Taya Acosta  
Vocal

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN – TACNA

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA EN INFORMÁTICA Y  
SISTEMAS


“USO DE REDES NEURONALES ARTIFICIALES PARA EL PRONÓSTICO DE  
DEMANDA DE HIELO INDUSTRIAL EN HIELOS SANTA LUCÍA S.R.L. - 2015”

TESIS SUSTENTADA Y APROBADA EL 18 DE DICIEMBRE DEL 2015  
ESTANDO EL JURADO CALIFICADOR INTEGRADO POR


Presidente :

  
Mgr. Erbert Francisco Osco Mamani


Secretario :

  
Ing. Gianfranco Alexey Málaga Tejada

Vocal :

  
MSc. Edgar Aurelio Taya Acosta

Asesor :

  
MSc. Edwin Antonio Hinojosa Ramos

## DEDICATORIA

*A Dios por permitirme llegar a este momento tan especial e importante en mi vida, con su infinita bondad y amor pude alcanzar triunfos inimaginables y vencer adversidades. A mis padres Fátima y Báxter, por ser mi guía, el pilar más importante, por demostrarme su cariño y por su apoyo incondicional. A mi tío Javier, a quien quiero como a un padre, por compartir momentos significativos y por estar siempre dispuesto a ayudarme. A mis amigos por su preocupación, por darme fuerza y por escucharme en todo momento.*

## **AGRADECIMIENTO**

A mis docentes universitarios, reconociendo el gran valor que tienen, la labor de enseñar y de formar los mejores profesionales. Un especial agradecimiento al Ing. Edwin Hinojosa Ramos, por su tiempo compartido, por impulsar la culminación de la presente tesis y que gracias a su apoyo hizo de esta experiencia una de las más especiales. Al MSc. Javier García Garrido de Gerona, España; por compartir su conocimiento de redes neuronales artificiales, por sus horas de sueño por el huso horario y por su erudición y entusiasmo en la docencia.

## RESUMEN

La predicción de la demanda de hielo industrial es de gran importancia hoy en día para las empresas pertenecientes al sector industrial en la toma de decisiones. El presente trabajo de investigación tuvo como objetivo predecir la demanda de hielo industrial en Hielos Santa Lucía S.R.L., utilizando redes neuronales artificiales, la cual se encuentra dentro de la Inteligencia Artificial Subsimbólica. El diseño de la investigación es no experimental. Para la muestra se ha seleccionado mediciones de volumen de producción en toneladas (t) agrupados por meses (enero 2010 – junio 2015) haciendo un total de sesenta y seis meses.

El desarrollo de la investigación comienza con la construcción del corazón matemático de las redes neuronales artificiales, luego se procede a la creación de propio código basándose en las fórmulas obtenidas y los pasos que componen el proceso de aprendizaje del perceptrón multicapa y el algoritmo Back Propagation.

Los resultados de la investigación apoyan la utilización de las redes neuronales artificiales como técnica confiable en la predicción de series de tiempo. Resuelve este tipo de problemas de manera eficiente, encontrándose resultados satisfactorios, con el valor de test de validación de 72% y el valor del error cuadrático medio de 10%.

*Palabras clave: red neuronal artificial, inteligencia artificial subsimbólica.*

**CONTENIDO**

DEDICATORIA	I
AGRADECIMIENTO	II
RESUMEN	III
CONTENIDO	IV
ÍNDICE DE TABLAS	VIII
ÍNDICE DE FIGURAS	IX
INTRODUCCIÓN	1
CAPÍTULO I: ESTUDIO METODOLÓGICO DEL PROBLEMA	2
1.1 Descripción del problema	2
1.2 Delimitación del problema	3
1.2.1 Alcances	3
1.2.2 Limitaciones	3
1.3 Planteamiento del problema	4
1.3.1 Problema general	4
1.3.2 Problemas específicos	4
1.4 Justificación del problema	4
1.5 Objetivos de la investigación	6
1.5.1 Objetivo general	6
1.5.2 Objetivos específicos	6

CAPÍTULO II: MARCO TEÓRICO, HIPÓTESIS Y VARIABLES	7
2.1 Antecedentes de la investigación	7
2.2 Fundamentos teóricos	9
2.2.1 Pronóstico	9
2.2.2 Demanda	9
2.2.3 Hielo industrial	10
2.2.4 Redes neuronales artificiales	11
2.3 Hipótesis	30
2.3.1 Hipótesis global	31
2.3.2 Subhipótesis	31
2.4 Variables e indicadores	32
2.4.1 Identificación de variables	32
2.4.2 Definición de las variables	32
2.5 Operacionalización de variables	33
2.5.1 Variable independiente	33
2.5.2 Variable dependiente	33
2.6 Diseño de la investigación	34
2.6.1 Diseño no experimental	34
2.6.2 Población y muestra	35
2.6.3 Técnicas e instrumentos para recolección de datos	35

2.6.4	Análisis de datos	36
2.6.5	Selección de pruebas estadísticas	36
CAPÍTULO III: DESARROLLO METODOLÓGICO DE LA INVESTIGACIÓN		37
3.1	Desarrollo de la Investigación	37
3.1.1	Descripción	37
3.1.2	Definición de variables	38
3.1.3	Selección de la topología	41
3.1.4	Normalización de datos	42
3.1.5	Fase de entrenamiento	43
3.1.6	Implementación	43
CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS		61
4.1	Presentación de los resultados obtenidos en la investigación	61
4.2	Análisis e interpretación de los resultados obtenidos	63
4.2.1	Ejecución de <code>preparacion_serie_temporal.m</code>	63
4.2.2	Ejecución de <code>programa1.m</code>	65
4.2.3	Ejecución de <code>uso1.m</code>	72
4.2.4	Predicción	74
4.2.5	Discusiones	75
CONCLUSIONES		76
RECOMENDACIONES		77

REFERENCIAS BIBLIOGRÁFICAS	78
ANEXOS	82

**ÍNDICE DE TABLAS**

Tabla 1.	Datos de la empresa.	38
Tabla 2.	Datos de entrenamiento.	38
Tabla 3.	Datos de validación.	39
Tabla 4.	Datos para el uso de la red.	40
Tabla 5.	Resumen de datos	44
Tabla 6.	Ejemplo de descenso del gradiente.	45
Tabla 7.	Resultado de los pesos por capa.	47
Tabla 8.	Ejemplo de función de escalamiento.	55
Tabla 9.	Datos de entrenamiento.	61
Tabla 10.	Datos de validación.	62
Tabla 11.	Datos para el uso de la red.	62
Tabla 12.	Predicción de demanda.	74

**ÍNDICE DE FIGURAS**

Figura 1.	Modelo de neurona McCulloch y Pitts	12
Figura 2.	Procesamiento de símbolos y procesamiento subsimbólico.	13
Figura 3.	Diagrama esquemático de una neurona.	14
Figura 4.	Motivaciones biológicas de las redes neuronales (a) Neuroanatomía de animales vivos (b) Conexiones de una neurona artificial.	15
Figura 5.	Modelo simple de aprendizaje automático.	18
Figura 6.	La función de transferencia sigmoide.	21
Figura 7.	Topología de la red (a) Un grafo acíclico y (b) un gráfico cíclico. El ciclo en (b) se enfatiza con líneas gruesas.	22
Figura 8.	Red de capa individual con una salida y dos entradas.	23
Figura 9.	Arquitectura del perceptron Multicapa.	24
Figura 10.	Serie de tiempo de los datos de entrenamiento.	39
Figura 11.	Serie de tiempo de los datos de validación.	40
Figura 12.	Serie de tiempo de los datos para el uso de la red.	41
Figura 13.	Esquema de una red multicapa.	42
Figura 14.	Demostración de rango.	45

Figura 15.	La red con pesos como voltímetros.	46
Figura 16.	La red como una caja negra.	47
Figura 17.	Comportamiento de una red.	48
Figura 18.	Ejemplo simple de arquitectura del Perceptrón.	50
Figura 19.	Ejemplo avanzado de arquitectura del Perceptrón.	51
Figura 20.	Ejemplo de aplicación de Pitágoras.	53
Figura 21.	Ejemplo de escalamiento.	56
Figura 22.	Variables y valores de <i>preparacion_serie_temporal.m</i> .	63
Figura 23.	Gráfico de líneas 2D utilizando ejes lineales de <i>serie</i> .	64
Figura 24.	Gráfico de líneas 2D utilizando ejes lineales de <i>patron</i> .	64
Figura 25.	Variables y valores de <i>programa1.m</i> .	66
Figura 26.	Gráfico de líneas 2D utilizando ejes lineales de <i>INPUT</i> .	67
Figura 27.	Gráfico de líneas 2D utilizando ejes lineales de <i>OUTPUT</i> .	67
Figura 28.	Gráfico de líneas 2D utilizando ejes lineales de <i>input</i> .	68
Figura 29.	Gráfico de líneas 2D utilizando ejes lineales de <i>output</i> .	68
Figura 30.	Gráfico de líneas 2D utilizando ejes lineales de <i>w</i> .	69
Figura 31.	Gráfico de líneas 2D utilizando ejes lineales de <i>cambio_peso</i> .	69

Figura 32.	Gráfico de líneas 2D utilizando ejes lineales de <i>e</i> .	70
Figura 33.	Gráfico de líneas 2D utilizando ejes lineales de <i>e_media</i> .	70
Figura 34.	Variables y valores de <i>uso1.m</i> .	72
Figura 35.	Gráfico de líneas 2D utilizando ejes lineales de <i>serie</i> .	73
Figura 36.	Gráfico de líneas 2D utilizando ejes lineales de <i>w</i> .	73

## INTRODUCCIÓN

La planeación en toda industria es una necesidad. Se puede decir incluso que el objetivo importante de la planeación es tratar de prever lo que sucederá en el futuro en base de una recopilación de hechos o sucesos acaecidos con anterioridad, esto repercute en la toma de decisiones y en la planificación de recursos para una mayor y eficiente producción. La predicción debe ser lo más realista posible y además ser confiable. Las redes neuronales artificiales son las que actualmente están causando un mayor impacto, debido a su extraordinaria aplicabilidad práctica.

El presente trabajo de investigación se ha estructurado de la siguiente manera: en el Capítulo I se describe la problemática con la que planteamos los objetivos e hipótesis; en el Capítulo II se integran las bases teóricas, definición de términos que se toman en cuenta para el desarrollo de la investigación; en el Capítulo III se elabora el desarrollo del proceso de aprendizaje de la red neuronal artificial para la predicción de demanda de hielo industrial; en el Capítulo IV se muestran los resultados obtenidos a partir de la validación de las hipótesis, en el Capítulo V se discuten los resultados obtenidos teniendo como soporte y análisis los antecedentes del presente estudio. Finalmente damos a conocer las conclusiones a las que se arribó en la presente investigación, luego se presentan las recomendaciones y la bibliografía que se consideró para el presente trabajo.

## **CAPÍTULO I**

### **ESTUDIO METODOLÓGICO DEL PROBLEMA**

#### **1.1 Descripción del problema**

Hielos Santa Lucía S.R.L. (2015) presentó una preocupación en saber qué volumen de producción realizar periódicamente con la finalidad de abastecer su almacén para poder vender en días de alta demanda, qué número de trabajadores debe tener para brindar un servicio rápido al cliente, entre otros factores. Luego de analizar, se determinó que el principal problema se debía a una falta de predicción, en ocasiones se encontraron casos como escasa agua para completar producción, niveles altos de facturación eléctrica, falta de personal y otros. Detectado el problema, la gerencia necesitaba un modelo de predicción, para tener tiempo suficiente para tomar decisiones a corto plazo y programar sus actividades.

Si se mantenía el problema generaría deficiencias en las áreas de producción y comercialización. Al estar en un sector pesquero, de productores marinos, plantas de procesamiento y empacadoras, en la temporada alta provoca un incremento de la demanda de hielo industrial y por este motivo debe aumentar el volumen de producción. Los niveles altos de energía eléctrica significaban un costo excesivo al no realizar una producción paulatinamente.

Prever lo que sucederá en el futuro en base de una recopilación de un hecho o suceso acaecido con anterioridad, es importante para la planeación en toda industria. A fin de dar una solución al problema, es conveniente realizar un pronóstico de demanda de hielo industrial. Por lo tanto, es necesario elegir un método eficaz para lo cual ha de desarrollarse un modelo de red neuronal artificial; una de sus aplicaciones es la predicción de series de tiempo, debido a sus óptimos resultados, requiere información cuantitativa el cual brinde una predicción de valores futuros del volumen de producción a través del tiempo, pretendiendo mejorar significativamente la toma de decisiones.

## **1.2 Delimitación del problema**

### **1.2.1 Alcances**

Para realizar este trabajo se dispuso de un conjunto de datos que corresponden al registro de ventas de hielo industrial, se construyó una red neuronal artificial utilizada para el pronóstico de la demanda de hielo industrial con el propósito de demostrar que dicha herramienta resuelve este tipo de problemas de manera eficiente, encontrándose resultados altamente satisfactorios.

### **1.2.2 Limitaciones**

No existieron limitaciones para efectuar el entrenamiento de una Red Neuronal Artificial para el pronóstico de demanda de hielo industrial en Hielos Santa Lucía S.R.L.

### **1.3 Planteamiento del problema**

#### **1.3.1 Problema general**

¿Es factible utilizar redes neuronales artificiales para predecir la demanda de hielo industrial en Hielos Santa Lucía S.R.L.?

#### **1.3.2 Problemas específicos**

- ¿Es realizable diseñar el modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.?
- ¿Es posible calcular valores futuros a partir del modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.?

### **1.4 Justificación del problema**

La elección e implementación de un método adecuado de pronósticos siempre ha sido un tema de gran importancia para las empresas. Un error significativo en el pronóstico de la demanda de hielo industrial podría generar a una empresa una pérdida monetaria demasiado grande y en un pronóstico erróneo aumentaría los gastos para la empresa.

La estimación del comportamiento futuro de algunas variables puede realizarse utilizando diversos métodos de pronóstico. Cada uno de los métodos de proyección tiene una aplicación de carácter especial que hace de su selección un problema de decisión influido por diversos factores, por ejemplo, la validez y disponibilidad de

los datos históricos, la precisión deseada del pronóstico, los beneficios de los resultados, los periodos futuros que se desee pronosticar y el tiempo disponible para hacer el estudio entre otros.

Existen diversos métodos para proyectar la demanda de hielo industrial. Dentro de los mismos se encuentra el método de las redes neuronales artificiales. Debido a su constitución y a sus fundamentos, presentan un gran número de características semejantes a las del cerebro humano. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos y de abstraer características esenciales. Las redes neuronales artificiales son metodologías novedosas que permiten hacer pronósticos donde hay cierto comportamiento no lineal y prometen un buen desempeño en este campo de investigación.

La principal ventaja de las aplicaciones de las redes neuronales reside en el procesado paralelo, adaptativo y no lineal. En la actualidad las redes neuronales artificiales son un campo muy consolidado en el que continuamente se desarrollan nuevos métodos y aplicaciones.

Aplicando las redes neuronales artificiales, podremos predecir la demanda de hielo industrial en distintos instantes, las cuales permitirán realizar cálculos complejos (cálculos empíricos), y así obtener valores de simulaciones de las cantidades estimadas con la que se contará en un futuro.

## **1.5 Objetivos de la investigación**

### **1.5.1 Objetivo general**

Predecir la demanda de hielo industrial en Hielos Santa Lucía S.R.L., utilizando redes neuronales artificiales.

### **1.5.2 Objetivos específicos**

- Diseñar el modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.
- Calcular valores futuros a partir del modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.

## **CAPÍTULO II**

### **MARCO TEÓRICO, HIPÓTESIS Y VARIABLES**

#### **2.1 Antecedentes de la investigación**

La demanda se establece como un requisito primordial que debe soportar el plan financiero o plan de ventas con lo que, la mayor parte de las veces, el proceso de planificación de la demanda está fuertemente condicionado por elementos que, en un principio, tiene que ver con el mercado y los clientes.

El método usual de planeación al interior de las empresas se inicia con un proceso de pronóstico de la demanda, pronóstico bien sea elaborado con un simple crecimiento sobre resultados anteriores o mediante herramientas tecnológicas que permiten efectuar innumerables cálculos para tratar de que el pronóstico sea lo más cercano a la realidad. Muchas de estas herramientas son extremadamente costosas y, peor aún, no son muchas las personas que pueden efectuar un pronóstico seguro a partir de los resultados de las mismas.

Actualmente, las redes neuronales artificiales son una de las técnicas más usadas para la resolución de pronósticos de este tipo debido a sus óptimos resultados y su fácil metodología de resolución.

La idea de utilizar la red neuronal artificial en el pronóstico de series de tiempo que fue aplicada por primera vez en 1964, cuando se utilizó una red neuronal artificial lineal para el pronóstico del clima (Sebastian, Verónica, Bibiana, & Ramon, 2003).

Se pueden encontrar muchos trabajos de investigación sobre este tema, a nivel internacional como el de Gómez (2010) denominado “Pronóstico de una serie temporal usando redes neuronales”, que tuvo como objetivo principal realizar la predicción y la comparación un conjunto de datos, los cuales corresponden a las demandas máximas mensuales de energía eléctrica proporcionados por la Comisión Federal de Electricidad, utilizando dos enfoques para lograr tal objetivo, la metodología de Box y Jenkins (1973) para series de tiempo y el uso de las redes neuronales artificiales.

A nivel nacional como el de Jiménez (2013) denominado pronóstico de demanda de llamadas en los *call center*, utilizando redes neuronales artificiales, que construyó una red neuronal artificial utilizada para el pronóstico de demanda de llamadas del centro de atención telefónica (*call center*) de clientes de la empresa ABC con el propósito de demostrar que dicha herramienta resuelve este tipo de problemas de manera eficiente, encontrando resultados altamente satisfactorios.

A nivel regional como el de Yanarico (2013) denominado desarrollo de un software utilizando redes neuronales artificiales para la simulación del ciclo

hidrológico de la laguna Aricota, que tuvo como objetivo desarrollar un software que permitiera realizar análisis de cada variable hidrológica, el entrenamiento de una red neuronal artificial (para obtener valores de incremento del volumen y nivel de agua de la laguna Aricota), y la simulación del ciclo hidrológico de la laguna Aricota.

## **2.2 Fundamentos teóricos**

### **2.2.1 Pronóstico**

Pronosticar es el arte de especificar información significativa acerca del futuro. Se menciona que los pronósticos jamás son perfectos debido a que, básicamente, se utilizan métodos que generan pronósticos sobre la base de la información previa. Los pronósticos serán menos confiables mientras mayor sea el horizonte que se va pronosticar. (Mendoza, 2011)

### **2.2.2 Demanda**

La demanda es el deseo que se tiene de un determinado producto pero que está respaldado por una capacidad de pago. Se refiere a las cantidades de un producto que los consumidores están dispuestos a comprar a los posibles precios del mercado. La demanda es la cantidad de bienes o servicios que el comprador o consumidor está dispuesto a adquirir a un precio dado y en un lugar establecido, con cuyo uso pueda satisfacer parcial o totalmente sus necesidades particulares o pueda tener acceso a su utilidad intrínseca. (Armstrong & Green, 2006)

### **2.2.3 Hielo industrial**

El hielo es agua congelada, su temperatura es siempre inferior a cero grados, pues el agua pura se congela a 0°C a presión atmosférica. A presiones mayores el punto de congelamiento disminuye levemente, así como también cuando el agua contiene sales en solución. (Iriando, 2007)

La fabricación de hielo comprende establecimientos que a través de procedimientos como purificación de agua y vaciado en moldes de diferentes formas y tamaños elaboran este producto por medio de congelación (Universidad de Texas, 2001). La fabricación comercial de hielo en bloques comenzó en 1869; consiste en rellenar moldes de metal con agua y sumergirlos en un baño de salmuera (generalmente cloruro sódico o cálcico) refrigerada a una temperatura muy inferior a la de congelación del agua. Tras varias horas, el agua se congela y los bloques de hielo se sacan de los moldes tras liberarlos por inmersión en agua; finalmente, se almacenan. (Departamento de Pesca y Acuicultura)

La fabricación de hielo industrial consiste en rellenar moldes de metal con agua y sumergirlos en un baño de salmuera refrigerada a una temperatura muy inferior a la de congelación del agua (ver anexo 2).

## **2.2.4 Redes neuronales artificiales**

### **2.2.4.1 Historia**

En 1943, McCulloch y Pitts demostraron que simples bloques similares a neuronas eran capaces de llevar a cabo toda la lógica posible de operaciones (ver figura 1). En ese tiempo Von Neumann y Turing discutieron aspectos interesantes de la naturaleza estadística y robusta del cerebro, como procesamiento de la información, pero fue sólo en la década de 1950 que las implementaciones de hardware reales de este tipo de redes comenzaron a ser producidas. El defensor más entusiasta de las nuevas máquinas de aprendizaje era Frank Rosenblatt que inventó la máquina del Perceptrón, que era capaz de realizar sencilla clasificación de patrones. Sin embargo, se manifestó que las nuevas máquinas de aprendizaje eran incapaces de resolver ciertos problemas, y en 1969 Minsky y Papert escribieron un concepto definitivo del Perceptrón, demostrando que en las redes había limitaciones. El núcleo del argumento contra las redes de ese momento se puede encontrar en la incapacidad de definición para resolver problemas XOR. El entusiasmo por las redes neuronales artificiales disminuyó hasta mediados de los ochentas, la primera vez que John Hopfield, un físico, analizó una clase particular de las redes neuronales artificiales cuando Rumelhart, McClelland y el grupo de PDP redescubrieron poderosas reglas de aprendizaje que trascendían la limitación discutida por Minsky y Papert. Durante la última década, el tema ha recibido una atención creciente. (Fyfe, 2000)

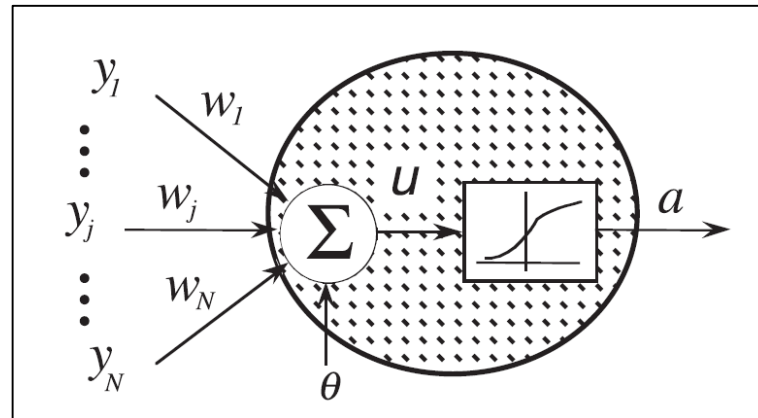


Figura 1. Modelo de neurona McCulloch y Pitts.

Fuente: Hu, Y. H., & Hwang, J.-N. (2002). Handbook of Neural Network Signal Processing. U.S.: CRC Press LLC.

#### 2.2.4.2 Inteligencia Artificial

Dentro de la Inteligencia Artificial se pueden distinguir dos grandes áreas, la Inteligencia Artificial Simbólica y la Subsimbólica (ver figura 2). La Subsimbólica trata de estudiar los mecanismos de los sistemas nerviosos, del cerebro, así como su estructura, funcionamiento y características lógicas, con la intención de diseñar programas basados en dichas características que se adapten y generen sistemas capaces de resolver problemas. En este caso el diseño es de abajo hacia arriba, ya que los sistemas diseñados son simples e idénticos, recogen las características físicas de los sistemas que tratan de imitar, y se van generando cómputos cada vez más complejos, de forma automática, mediante mecanismo prefijados de aprendizaje. Es en este campo donde se encuentran las redes de neuronas artificiales. (Isasi Viñuela & Galván León, 2004)

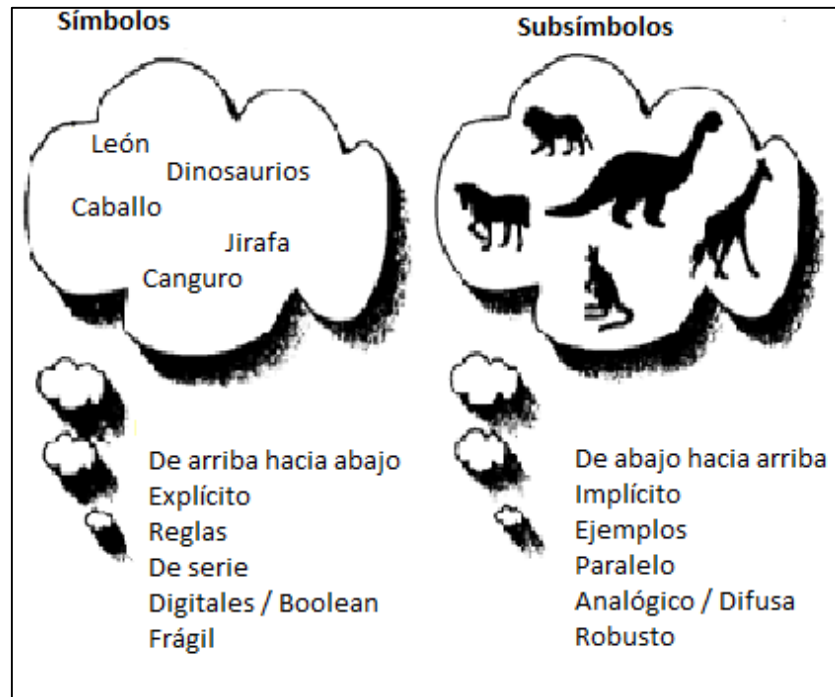


Figura 2. Procesamiento de símbolos y procesamiento subsimbólico.

Fuente: Bigus, J. P. (1996). Data Mining with Neural Networks. New York: McGraw-Hill.

### 2.2.4.3 Definición

Una red neuronal artificial es un sistema de procesamiento de información que tiene ciertas características de rendimiento en común con las redes neuronales biológicas (ver figura 3). Se han desarrollado como generalizaciones de modelos matemáticos de la cognición humana. (Fausett, 1994)

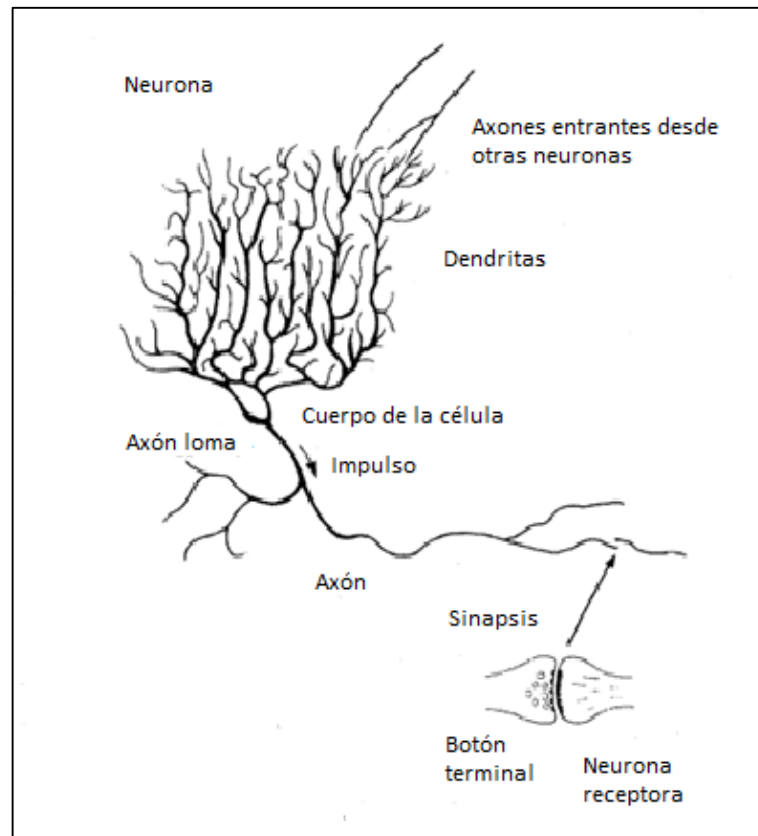


Figura 3. Diagrama esquemático de una neurona.

Fuente: Zurada, Z. M. (1992). Introduction to Artificial Neural Systems. Saint Paul, Minnesota: West.

#### 2.2.4.4 Inspiración

Se inspiran en los procesos biológicos de tratamiento de la información, incluyendo específicamente el sistema nervioso y su unidad básica: la neurona (ver figura 4). Las señales se propagan en forma de diferencias de potencial entre el interior y el exterior de las células. Las dendritas llevan señales de otras neuronas en el cuerpo celular o soma, posiblemente multiplicando cada señal entrante mediante un coeficiente de transferencia de ponderación. En el soma, célula

capacitancia integra las señales que recogen en el axón. Una vez la señal combinada supera un umbral determinado de células, una señal, el potencial de acción, se transmite a través del axón. Linealidades celulares hacen que la acción compuesta potencial de una función no lineal de la combinación de las señales que llegan. El axón se conecta a través de las sinapsis con las dendritas de las neuronas posteriores. La sinapsis opera a través de la descarga de neurotransmisores químicos en todos los agujeros intercelulares, y pueden ser excitatorio, que tiende a disparar la siguiente neurona o inhibitorio, que tiende a prevenir el disparo de la siguiente neurona. (Sarangapani, 2006)

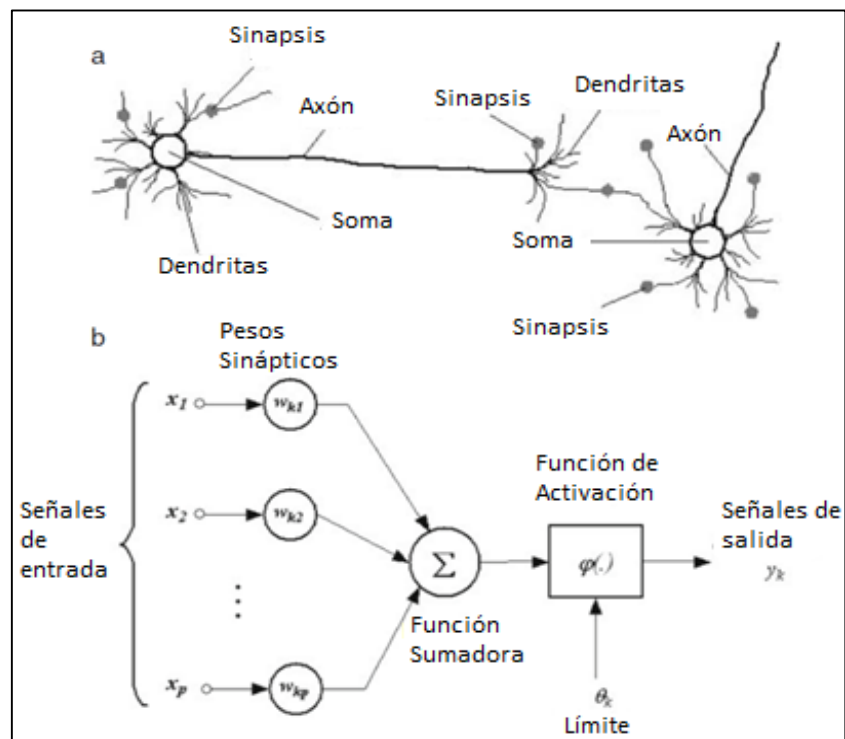


Figura 4. Motivaciones biológicas de las redes neuronales (a) Neuroanatomía de animales vivos (b) Conexiones de una neurona artificial.

Fuente: Yeung, D. S., Cloete, I., Shi, D., & Ng, W. W. (2010). Sensitivity Analysis for Neural Networks. New York: Springer-Verlag Berlin Heidelberg.

#### **2.2.4.5 Neurona**

Una neurona es una unidad de procesamiento de información que es fundamental para el funcionamiento de una red neuronal artificial. (Haykin, 1999)

#### **2.2.4.6 Retroalimentación**

Para la retroalimentación, se dice que existe un sistema dinámico donde la entrada y la salida dan lugar a uno o más caminos cerrados para la transmisión de las señales de todo el sistema. (Haykin, 1999)

#### **2.2.4.7 Sinapsis**

Una sinapsis se utiliza para conectar una capa a otra. Las sinapsis contienen las matrices de peso utilizados por la red neuronal. Las matrices de peso tienen los valores de conexión entre cada una de las neuronas en las dos capas que son conectado por esta sinapsis. (Heaton, 2010)

#### **2.2.4.8 Aprendizaje**

La importancia principal de una red neuronal es su capacidad para aprender de su entorno. En el contexto de las redes neuronales, el aprendizaje se define como un proceso por el cual los parámetros libres de una red neuronal se adaptan a través de un proceso continuo de la estimulación por el medio ambiente. El tipo de aprendizaje es determinado por la manera en que los cambios de parámetros se llevan a cabo (ver figura 5). La red es estimulada por el medio ambiente, los

cambios en la red como resultado de la estimulación y la red responde al ambiente en una nueva forma después de la ocurrencia del cambio. (Yeung, Cloete, Shi, & Ng, 2010)

Hemos encontrado que después de un número finito de pasos la regla de aprendizaje por refuerzo proporcionará una solución correcta si se cumplen ciertas condiciones. Sin embargo, hay otras reglas de aprendizaje que le darán soluciones correctas en determinadas condiciones. Estas reglas también se basan en los procedimientos de gradiente para minimizar ciertas funciones de criterio, como funciones de costo (por errores de clasificación), a veces llamadas funciones de error o funciones de energía. (Veelenturf, 1995)

Claramente una de las diferencias entre el aprendizaje automático y la estadística es la terminología, que por supuesto es una función de la historia de las disciplinas. Esto se ve agravado por el estándar en matemáticas. Otra diferencia fácilmente descubierto es que el aprendizaje de la máquina tiene mejores nombres para sus actividades. Las diferencias más importantes son que aprendizaje automático tiende a tener un énfasis en heurísticas simples y rápidas, mientras que las estadísticas tienden a comenzar con un modelo para los datos, a menudo no existe un modelo real de los datos o solo un trivial en el aprendizaje de la máquina. (Dunne, 2007)

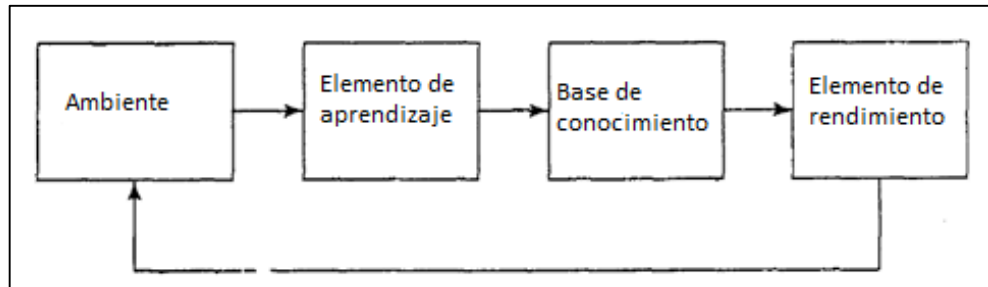


Figura 5. Modelo simple de aprendizaje automático.

Fuente: Haykin, S. (1999). *Neural Network: A Comprehensive Foundation*. Delhi: Prentice-Hall.

#### 2.2.4.9 Entrenamiento

La formación es el proceso por el cual estos pesos de conexión están asignados. Los pesos utilizados en las conexiones entre las diferentes capas tienen mucha importancia en el funcionamiento de la red neuronal y la caracterización de una red. (Rao & Rao, 1995)

La mayoría de algoritmos de entrenamiento comienzan mediante la asignación de números aleatorios a una matriz de ponderaciones. Entonces, se examina la validez de la red neuronal. Las ponderaciones se basan ajuntado de lo bien que la red neuronal realizó y la validez de los resultados. Este proceso se repite hasta que el error de validación está dentro de un límite aceptable. Hay muchas maneras de entrenar redes neuronales. Métodos de entrenamiento generalmente se dividen en las categorías de enfoques híbridos supervisadas, sin supervisión, y varios. (Heaton, 2008)

Al parecer, la primera descripción de un algoritmo para entrenar las redes Multicapa estaba contenida en la tesis de Paul Werbos en 1974. Esta tesis presenta el algoritmo en el contexto de las redes generales, con las redes neuronales como caso especial, y no se difundió en la comunidad de redes neuronales. No fue sino hasta mediados de la década de 1980 que el algoritmo Back Propagation fue redescubierto y ampliamente publicitado. Fue redescubierto independientemente por David Rumelhart, Geoffrey Hinton y Ronald Williams, David Parker y Yann Le Cun. El algoritmo fue popularizado por su inclusión en el grupo de procesamiento distribuido en paralelo libro dirigido por psicólogos David Rumelhart y James McClelland, la publicación de este libro impulsado un torrente de investigación en redes neuronales. El perceptrón multicapa, entrenado por el algoritmo Back Propagation, es actualmente la red neuronal más utilizada. (Hagan, Demuth, & Beale, 1996)

#### **2.2.4.10 Simulación**

La simulación es una técnica para estimar las medidas de desempeño del sistema modelado. Se consideran varias técnicas para la simulación de modelos de procesamiento, utilizando metodologías de programación convencionales. (Freeman & Skapura, 1991)

#### **2.2.4.11 Características**

Las redes neuronales artificiales se pueden caracterizar más adecuadamente como modelos computacionales con propiedades particulares, tales como la capacidad de adaptarse o aprender, generalizar, agrupar u organizar los datos y cuya operación se basa en el procesamiento en paralelo. (Krose & van der Smagt, 1996)

#### **2.2.4.12 Función de activación**

La función de activación  $f$  también se conoce como una función de aplastamiento (ver figura 6), que mantiene la salida entre ciertos límites, es el caso en la neurona biológica. Diferentes funciones están en uso, todos los cuales tienen por encima de la limitación de la propiedad, la función de activación más común es la función sigmoide que es una función continuamente diferenciable que satisface la relación. (Graupe, 2007)

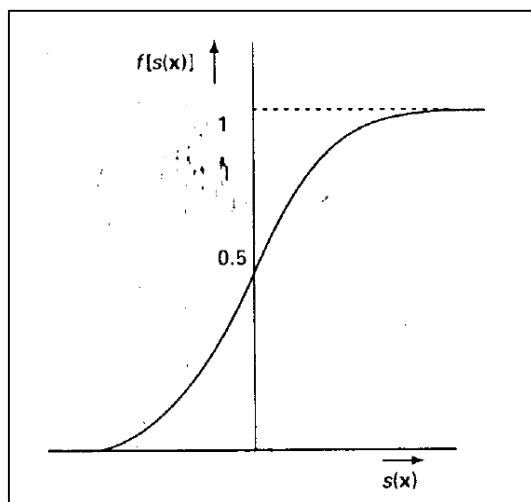


Figura 6. La función de transferencia sigmoide.

Fuente: Vealenturf, L. (1995). Analysis and Applications of Artificial Neural Networks. Reino Unido: Prentice-Hall.

#### 2.2.4.13 Funcionalidad

La neurona realiza una suma de las señales eléctricas al llegar a sus dendritas. Esta suma se compara con un umbral para determinar si la neurona se excita, dando como resultado una generación de indicio a la dendrita de otra neurona. A finales del siglo diecinueve al comienzo no se encontraron señales en una neurona a estar sujeto a la atenuación en las sinapsis, es decir, la sinapsis ayudó a controlar la fuerza de la señal eléctrica pasado a la neurona. (Taylor, 2006)

#### 2.2.4.14 Estructura

Las neuronas procesan la información en la membrana. La membrana regula tanto la transmisión y el procesamiento de la información. La suma de las señales y la comparación con un umbral es un combinado efecto de la membrana y el

citoplasma. Si se genera un impulso se transmiten y las sinapsis establecen algunos transmisores de moléculas libres. La neurona tiene dendritas, un cuerpo celular y un axón. Los mismos tres elementos estarán presentes en nuestras unidades de computación artificiales. (Rojas, 1996)

#### 2.2.4.15 Topología

En una red neuronal artificial, múltiples neuronas se interconectan para formar una red para facilitar la distribución informática. La configuración de las interconexiones se puede describir de manera eficiente con un grafo dirigido. Un grafo dirigido consta de nodos (en el caso de una red neuronal artificial, las neuronas, así como insumos externos) y arcos dirigidos (en el caso de una red neuronal artificial, enlaces sinápticos). La topología de la gráfica puede ser categorizado (ver figura 7) como sea acíclico o cíclico. (Hu & Hwang, 2002)

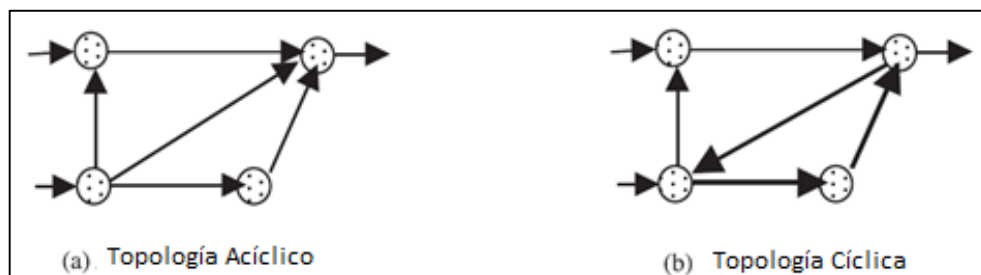


Figura 7. Topología de la red (a) Un grafo acíclico y (b) un gráfico cíclico. El ciclo en (b) se enfatiza con líneas gruesas.

Fuente: Hu, Y. H., & Hwang, J.-N. (2002). Handbook of Neural Network Signal Processing. U.S.: CRC Press LLC.

### 2.2.4.16 Arquitectura

Incluye definir el número de capas, el número de neuronas en cada capa, y el esquema de interconexión entre las neuronas. La selección del número de capas es controlada por el algoritmo de entrenamiento. Algunos algoritmos de entrenamiento pueden requerir solo una capa, mientras que otros pueden requerir un mínimo de tres capas. (Galushkin, 2007)

De manera que esencialmente el desarrollo podría ser caracterizado como un proceso de esculpir una red neuronal artificial de un número fijo de unidades y su azar interconexiones. (Quinlan, 2003)

Como ejemplo se tiene la arquitectura de una red de capa individual con una salida y dos entradas (ver figura 8) y la arquitectura del perceptrón multicapa (ver figura 9).

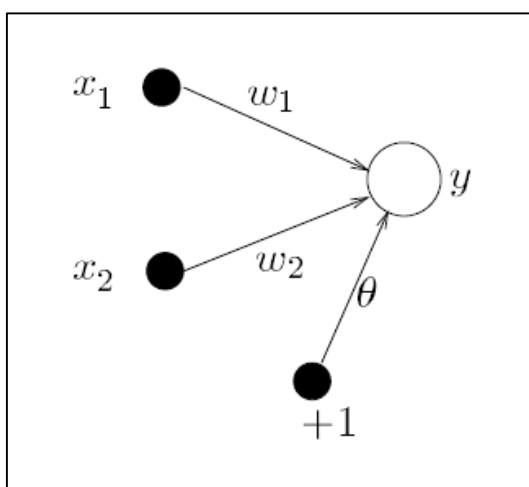


Figura 8. Red de capa individual con una salida y dos entradas.

Fuente: Krose, B., & van der Smagt, P. (1996). An introduction to neural networks. Amsterdam: University of Amsterdam.

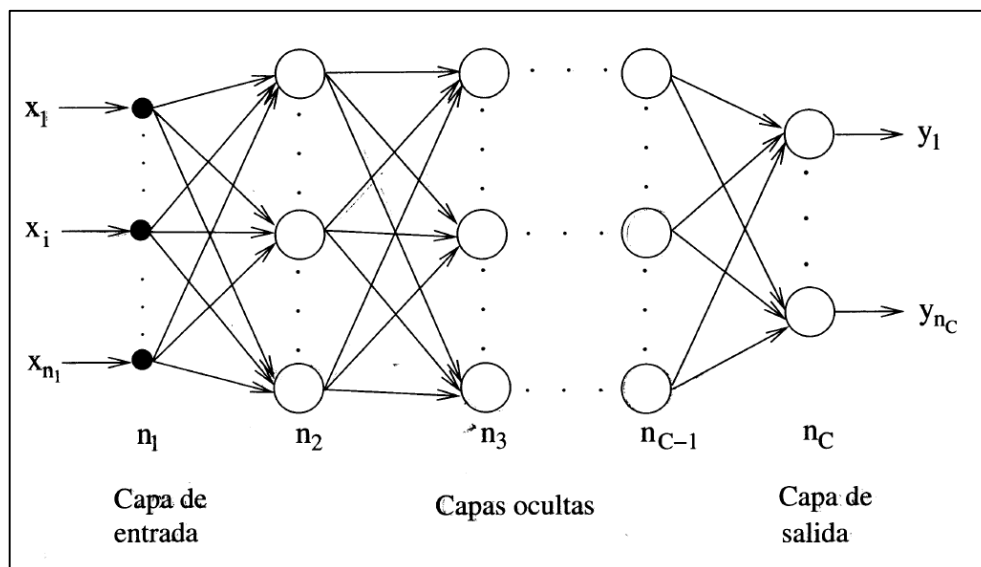


Figura 9. Arquitectura del perceptrón multicapa.

Fuente: Isasi Viñuela, P., & Galván León, I. M. (2004). *Redes de neuronas artificiales: Un enfoque práctico*. España: Pearson.

#### 2.2.4.17 Back Propagation

En el algoritmo de Back Propagation se requiere de una capa de entrada, una capa de salida y una capa oculta. Se selecciona el número de capas ocultas en base a la complejidad del problema. El número de neuronas de entrada y salida de cada capa es específico para un problema. Las interconexiones entre neuronas son controladas por el algoritmo de entrenamiento y la naturaleza del problema. (Khare & Nagendra Shiva, 2007)

#### 2.2.4.18 Enfoques

Galushkin (2007) considera que son posibles cuatro enfoques para la investigación de redes neuronales:

- Enfoque psicológico, cuando es necesario modelar algunos paradigmas psicológicos que requieran un desarrollo e investigación de la red neuronal con un poco de estructura definida.
- Enfoque neurofisiológico, cuando la red neuronal es desarrollada e investigada sobre la base de los conocimientos de la estructura de una parte del cerebro.
- Enfoque algorítmico, cuando se formula algún problema matemático y una red neuronal adecuada con el algoritmo correspondiente ajustado a esta solución está diseñado sobre la base de esta formulación.
- Enfoque sistemático que combina todos los enfoques mencionados anteriormente.

#### **2.2.4.19 Aplicaciones**

El uso de las redes neuronales en aplicaciones, incluyendo control de control de procesos, robótica, aplicaciones industriales y aeroespaciales industriales, entre otros, ha experimentado recientemente un rápido crecimiento. (Zurada, 1992)

Han logrado éxitos notables en el campo de los negocios de previsión. Sin embargo, es importante señalar que no puede ser una panacea para cada tarea de predicción en todas las circunstancias. Pronosticar competiciones sugieren que hay un método único, incluidas las redes neuronales, es universalmente el mejor para todos los tipos de problemas en cada situación. (Zhang, 2004)

#### **2.2.4.20 Problemas a resolver**

Hay muchos problemas diferentes que pueden ser resueltos con una red neuronal. Sin embargo, redes neuronales se utilizan comúnmente para tratar determinados tipos de problemas: clasificación, predicción, reconocimiento de patrones y optimización. (Heaton, 2008)

#### **2.2.4.21 Comparaciones**

La gran diferencia entre las redes neuronales artificiales y los modelos clásicos como ARIMA (Modelos Autorregresivos Integrados de Medias Móviles), ARCH (Heterocedasticidad Condicional Autorregresiva) y GARCH (Heteroscedasticidad Autorregresiva Condicional Generalizada) es la falta de hipótesis *a priori*. Es posible estudiar y hacer predicciones sobre cualquier conjunto de datos con las redes neuronales artificiales. La principal hipótesis de los algoritmos neuronales es que la relación no lineal que genera los datos pueden ser aprendidos. (Tirozzi, y otros, 2006)

#### **2.2.4.22 Limitaciones**

Las diferencias entre los modelos de Back Propagation y el modelo natural no son muy importante en sí mismos. El diseño de modelos artificiales no pretendía obtener una copia perfecta del modelo natural, sino una serie de conductas cuyo funcionamiento sea semejante tanto como sea posible. Sin embargo, una estrecha similitud entre ambos es indispensable para mejorar la producción y aumentar la

complejidad de la red neuronal artificial y puede resultar en más comportamientos inteligentes. Las redes neuronales artificiales se encuentran todavía en una fase de desarrollo y, posiblemente, incluso en su inicial fase. Su verdadero potencial está lejos de ser alcanzado, o incluso de sospecha. (Rabuñal & Dorado, 2006)

#### **2.2.4.23 Escalamiento**

La transformación implica la cuestión más general de escalar los datos para su presentación a la red neuronal. La mayoría de los modelos de redes neuronales aceptan datos numéricos solo en el rango de 0 a 1 ó -1 a 1, dependiendo de las funciones de activación utilizados en los elementos de procesamiento neuronal. En consecuencia, los datos por lo general deben ser escalados hacia abajo para ese rango puede ser directamente a la gama de 0 a 1. Si los valores de los datos son asimétricos, en una función logarítmica a nivel de pieza lineal o pueden ser utilizados para transformar, puede ser representado por tipos codificados con valores de 0 y 1, o pueden ser asignados valores en el rango continuo deseado. (Bigus, 1996)

#### **2.2.4.24 Perceptrón**

Rosenblatt (1958) define un perceptrón como una máquina que aprende, utilizando ejemplos, asigna los vectores de entrada (muestras) a diferentes clases, utilizando una función lineal de las entradas. Minsky y Papert (1969) describen el perceptrón como un gradiente de descenso estocástico algoritmo que intenta separar

linealmente un conjunto de datos de entrenamiento n-dimensionales. Se utiliza la palabra perceptrón en el primer sentido como una máquina, siguiendo Rosenblatt, y se refiere explícitamente al perceptrón como un algoritmo de aprendizaje, siempre que sea necesario. Un perceptrón puede ser representado por un solo nodo que aplica una función de paso a la suma ponderada neta de sus entradas. El patrón de entrada es considerado que pertenecen a una clase u otra, dependiendo de si la salida es 0 ó 1. (Mehrotra, Mohan, & Ranka, 1997)

El perceptrón utiliza una regla de corrección de errores para cambiar los pesos de cada unidad que hace que las respuestas erróneas a estímulos que presentan a la red, libre de bucles que tiene sus unidades dispuestas en capas, con cada unidad aportaciones solo para unidades en la capa siguiente de la secuencia. La primera capa comprende unidades de entrada; se puede entonces ser varias capas de unidades ocultas entrenables que llevan una representación interna, y por último está la capa de unidades de salida, también con pesos sinápticos entrenables. (Arbib, 2003)

#### **2.2.4.25 Pronosticar**

Hay muchos problemas de la vida real en la que los acontecimientos futuros deben predecirse sobre la base de la historia pasada. Una tarea de ejemplo es el de predecir el comportamiento de los índices bursátiles. Weigend y Huberman (1990) observan que la predicción se basa en dos tipos de conocimiento: conocimiento de

las leyes subyacentes, un medio de predicción muy potente y precisa, y el descubrimiento de fuertes regularidades empíricas en las observaciones de un sistema dado. Sin embargo, las leyes subyacen en el comportamiento de un sistema no son fácilmente descubiertas y regularidades empíricas o periodicidades no siempre son evidentes, a menudo pueden ser enmascarados por el ruido. Aunque la predicción perfecta casi nunca es posible, las redes neuronales se pueden utilizar para obtener razonablemente buenas predicciones en un número de casos. Por ejemplo, las redes neuronales han tenido éxito en el aprendizaje del ciclo de once años de datos de manchas. (Mehrotra, Mohan, & Ranka, 1997)

#### **2.2.4.26 Back Propagation**

El algoritmo de Back Propagation, basado en la regla delta generalizada, es seleccionado para entrenar la red multicapa de alimentación directa. La entrada se propaga hacia adelante, desde la capa de entrada a través la capa media a la capa de salida, y luego la salida resultante se propaga hacia atrás, desde la capa de salida a través de la zona central hacia la capa de entrada, para calcular el error. Posteriormente, los derivados de error de peso y los cambios de peso se calculan. El entrenamiento de la red consiste en presentar repetidamente los patrones en la formación establecidas en la red hasta que el error generado en la inicial aguda de la pendiente de la curva y su sedimentación lenta indica que la red tenía ninguna dificultad en el aprendizaje de los patrones de entrada y en la formación. Se completan el número de iteraciones. Es decir, los patrones en el conjunto de

entrenamiento son presentados a la red n-veces antes de que la red podría aprender a asociar cada entrada patrón con su patrón de salida correspondiente. (Parks, Levine, & Long, 1998)

La red neuronal Multicapa es la red neuronal que más se aplica, que se ha utilizado en la mayoría de las investigaciones hasta el momento. Un algoritmo de Back Propagation se puede utilizar para entrenar a estas redes de alimentación directa multicapa con un diferenciable al transferir funciones para llevar a cabo la función de aproximación, la asociación de patrón y clasificación de patrones. Back Propagation, este término se refiere al proceso de por el cual los derivados de error en la red, con respecto a los pesos de la red y sesgos, se pueden calcular. La formación de las redes neuronales artificiales por propagación hacia atrás implica tres etapas, según Fausett (1994) son: Primera etapa, la alimentación hacia adelante del patrón de entrenamiento de entrada; segunda etapa, el cálculo y la propagación hacia atrás del error asociado; tercera etapa, el ajuste de los pesos. Este proceso se puede utilizar con un número de diferentes estrategias de optimización. (Sapuan & Mujtaba, 2010)

### **2.3 Hipótesis**

Hipótesis descriptivas de un dato o valor que se pronostica, se utilizan a veces en estudios descriptivos, para intentar predecir un dato o valor en una o más variables que se van a medir u observar. Pero cabe comentar que no en todas las

investigaciones descriptivas se formulan hipótesis de esta clase o que sean afirmaciones más generales. No es sencillo realizar estimaciones con relativa precisión con respecto a ciertos fenómenos. (Hernández Sampieri, Fernández Collado, & Baptista Lucio, Metodología de la Investigación, 2010)

### **2.3.1 Hipótesis global**

**H<sub>0</sub>:** Las redes neuronales artificiales no predicen la demanda de hielo industrial en Hielos Santa Lucía S.R.L.

**H<sub>1</sub>:** Las redes neuronales artificiales predicen la demanda de hielo industrial en Hielos Santa Lucía S.R.L.

### **2.3.2 Subhipótesis**

#### **2.3.2.1 Primera Subhipótesis**

**H<sub>0</sub>:** La validación de la predicción de demanda de hielo industrial con el uso de redes neuronales, no será superior a 55%.

**H<sub>1</sub>:** La validación de la predicción de demanda de hielo industrial con el uso de redes neuronales, será superior a 55%.

#### **2.3.2.2 Segunda Subhipótesis**

**H<sub>0</sub>:** El error de entrenamiento del modelo de predicción de demanda de hielo industrial con el uso de redes neuronales, no será inferior a 15%.

**H<sub>1</sub>:** El error de entrenamiento del modelo de predicción de demanda de hielo industrial con el uso de redes neuronales, será inferior a 15%.

## **2.4 Variables e indicadores**

### **2.4.1 Identificación de variables**

*Variable independiente:* Redes neuronales artificiales.

*Variable dependiente:* Demanda de hielo industrial.

### **2.4.2 Definición de las variables**

#### **a. Redes neuronales artificiales**

*Por la función que cumplen en la hipótesis:* Variable independiente.

*Por su naturaleza:* Desactiva.

*Por el método de estudio:* Cualitativo.

*Por la posesión de la característica:* Discreta.

*Por los valores que adquiere:* Politómica.

#### **b. Demanda de hielo industrial**

*Por la función que cumplen en la hipótesis:* Variable dependiente.

*Por su naturaleza:* Activa.

*Por el método de estudio:* Cuantitativo.

*Por la posesión de la característica:* Continuo.

*Por los valores que adquiere:* Politómica.

## **2.5 Operacionalización de variables**

### **2.5.1 Variable independiente**

*Variable independiente:* Redes neuronales artificiales.

*Definición conceptual:* Las redes neuronales artificiales son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso.

*Definición operacional:* Para tener una red neuronal la cual se ajuste de manera adecuada a los datos, se tiene que realizar varias configuraciones como el número de capas ocultas, el número de neuronas por cada capa y las funciones de transferencias.

*Indicadores:* Valor de validación (aceptación de resultado obtenido por la red), valor del error cuadrático medio (error de entrenamiento de la red).

*Valores:* Numéricos (expresado en porcentaje).

### **2.5.2 Variable dependiente**

*Variable dependiente:* Demanda de hielo industrial.

*Definición conceptual:* La demanda es la cantidad de hielo industrial que puede ser adquirida por consumidores, determinada en función del volumen de producción.

*Definición operacional:* Para realizar la predicción usando series de tiempo el cual es necesario que el conjunto de datos cumpla cierta hipótesis para que, en primer lugar, se busque un modelo estadístico que se ajuste a estos de manera adecuada y posteriormente realizar la predicción.

*Indicadores:* Cantidad (el número total de los datos), valor máximo (el mayor número de los datos), valor mínimo (el menor número de los datos).

*Valores:* Numéricos.

## **2.6 Diseño de la investigación**

### **2.6.1 Diseño no experimental**

La presente investigación está enfocado en un diseño no experimental, de investigación transeccional descriptivo. Tiene como objetivo indagar la incidencia de las modalidades o niveles de una o más variables en una población. Es un estudio puramente descriptivo y cuando establecen hipótesis, éstas son también descriptivas de pronóstico de valores. (Hernández Sampieri, Fernández Collado, & Baptista Lucio, Metodología de la Investigación, 2010)

### **2.6.2 Población y muestra**

*Población:* Se ha seleccionado mediciones de volumen de producción en toneladas (t) agrupados por meses (enero 2010 - junio 2015), haciendo un total de sesenta y seis meses.

*Muestra:* Debido a que los datos están almacenados y son manipulados electrónicamente, no fue necesario de establecer una muestra, fue conveniente trabajar con toda la población, por lo tanto, hablamos de una población muestral.

### **2.6.3 Técnicas e instrumentos para recolección de datos**

La recopilación de datos es importante para establecer las bases de parámetros para el entrenamiento de la red neuronal artificial y la búsqueda de diferente información documentaria que guarde relación con la investigación.

*Métodos Pronóstico:* Método de diseño de pronóstico con redes neuronales artificiales.

*Técnicas:* Algoritmo Back Propagation – Perceptrón multicapa.

*Herramientas:* La hoja de cálculo de Microsoft Excel 2013 y el software matemático MATLAB 8.6.0.

*Fuente:* Los datos empleados fueron obtenidos de la empresa Hielos Santa Lucía S.R.L., la cual proporcionó archivos históricos de datos reales y de gran utilidad para el propósito de la investigación.

#### **2.6.4 Análisis de datos**

Para la implementación de las redes neuronales artificiales para el pronóstico de demanda de hielo industrial, no fue necesario realizar el pre-procesamiento de datos del volumen de producción (t), dado que los datos se encontraron libres de ruido (no contiene valores nulos ni valores con formato inconsistente). La confiabilidad del resultado estuvo dada por el valor del test validación y del error cuadrático medio.

#### **2.6.5 Selección de pruebas estadísticas**

Por el diseño de la investigación utilizado no hubo pruebas estadísticas. Los resultados obtenidos de la red neuronal artificial sirvieron para verificar el porcentaje del test de validación y del error cuadrático medio.

## **CAPÍTULO III**

### **DESARROLLO METODOLÓGICO DE LA INVESTIGACIÓN**

#### **3.1 Desarrollo de la Investigación**

##### **3.1.1 Descripción**

Hielos Santa Lucia S.R.L. se dedica a la fabricación de hielo industrial, ver datos en tabla 1, con una presentación en bloques de 50 kl. La ventaja de estos son el almacenamiento es prolongado al ser compacto, facilita la manipulación y transporte y puede ser reducido de tamaño. Es una planta que tiene una capacidad de 40 t/d. Para la operación llevan un control de seguridad y un control de seguridad. Los bloques son almacenados en cámaras frigoríficas. Ofrece un servicio de moladora de hielo de 15 t/h de capacidad.

Su misión es alcanzar el liderazgo en el sector industrial y expandir el negocio a nivel nacional. Su visión es satisfacer integralmente a sus clientes aplicando políticas de calidad, oportunidad y buen servicio. Sus principales clientes son de la industria pesquera, que lo utilizan para conservación de pescado o diversos tipos.

Para el desarrollo de la red neuronal artificial, se utilizó la hoja de cálculo Excel y la herramienta de software matemático MATLAB.

Tabla 1. Datos de la empresa.

RUC	20519798345
Razón Social	Hielos Santa Lucía S.R.L.
Teléfono	(053) 481164
Dirección legal	Mza. F Lote. 01 Parque Industrial
Distrito	Ilo
Ciudad	Ilo
Departamento	Moquegua
Estado	Activo
CIIU	74996

Fuente: Hielos Santa Lucía S.R.L.

### 3.1.2 Definición de variables

Las variables de entrada de la red neuronal constituyen los valores de la serie temporal, para lo cual se ha estructurado la información en tres partes:

- De enero 2010 a enero 2012: Datos iniciales de entrenamiento (ver tabla 2).

Tabla 2. Datos de entrenamiento.

	Mes	Volumen de producción (t)		
		2010	2011	2012
Datos para la fase de entrenamiento	<i>Enero</i>	2045,56	1879,10	2141,03
	<i>Febrero</i>	1226,11	1750,52	-
	<i>Marzo</i>	785,42	1618,00	-
	<i>Abril</i>	368,85	1111,07	-
	<i>Mayo</i>	343,49	1057,30	-
	<i>Junio</i>	340,95	1015,60	-
	<i>Julio</i>	266,50	521,45	-
	<i>Agosto</i>	644,52	115,20	-
	<i>Septiembre</i>	720,95	105,80	-
	<i>Octubre</i>	663,99	368,25	-
	<i>Noviembre</i>	1745,30	694,42	-
	<i>Diciembre</i>	1484,55	1765,95	-

Fuente: Hielos Santa Lucía S.R.L.

A continuación se grafica la serie de tiempo de los datos para la fase de entrenamiento (ver figura 10).

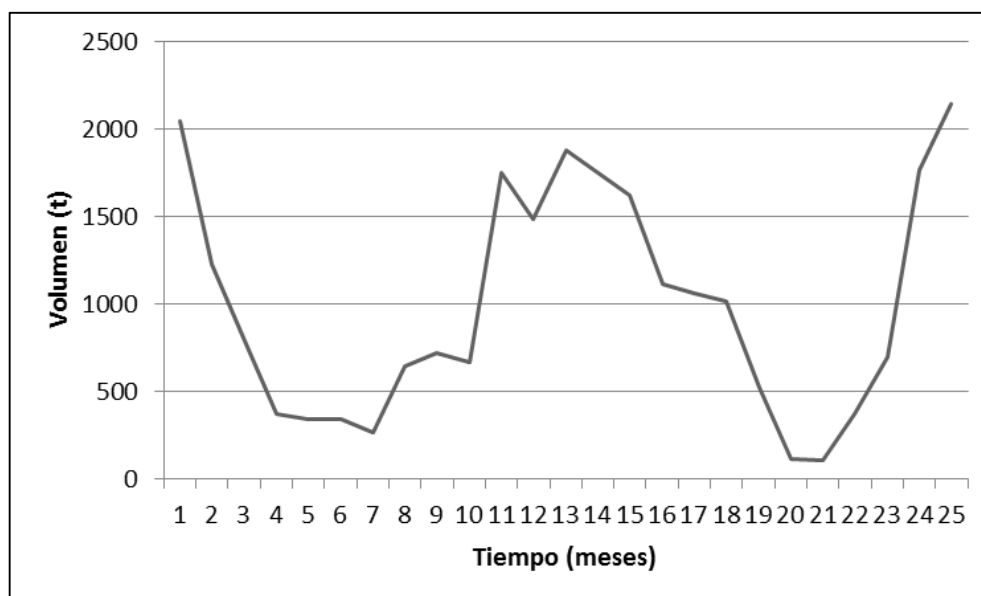


Figura 10. Serie de tiempo de los datos de entrenamiento.

Fuente: Elaboración propia.

- De febrero 2012 a febrero 2014: Datos de validación (ver tabla 3).

Tabla 3. Datos de validación.

Mes	Volumen de producción (t)		
	2012	2013	2014
<i>Enero</i>	-	1729,31	2106,11
<i>Febrero</i>	965,62	1130,04	1194,87
<i>Marzo</i>	925,80	447,29	-
<i>Abril</i>	585,30	298,77	-
<i>Mayo</i>	693,50	545,60	-
<i>Junio</i>	689,70	785,90	-
<i>Julio</i>	281,50	397,55	-
<i>Agosto</i>	78,75	219,00	-
<i>Septiembre</i>	67,40	233,35	-
<i>Octubre</i>	324,25	586,27	-
<i>Noviembre</i>	811,75	1953,40	-
<i>Diciembre</i>	1390,65	2244,13	-

Fuente: Hielos Santa Lucía S.R.L.

A continuación se grafica la serie de tiempo de los datos para la fase de validación (ver figura 11).

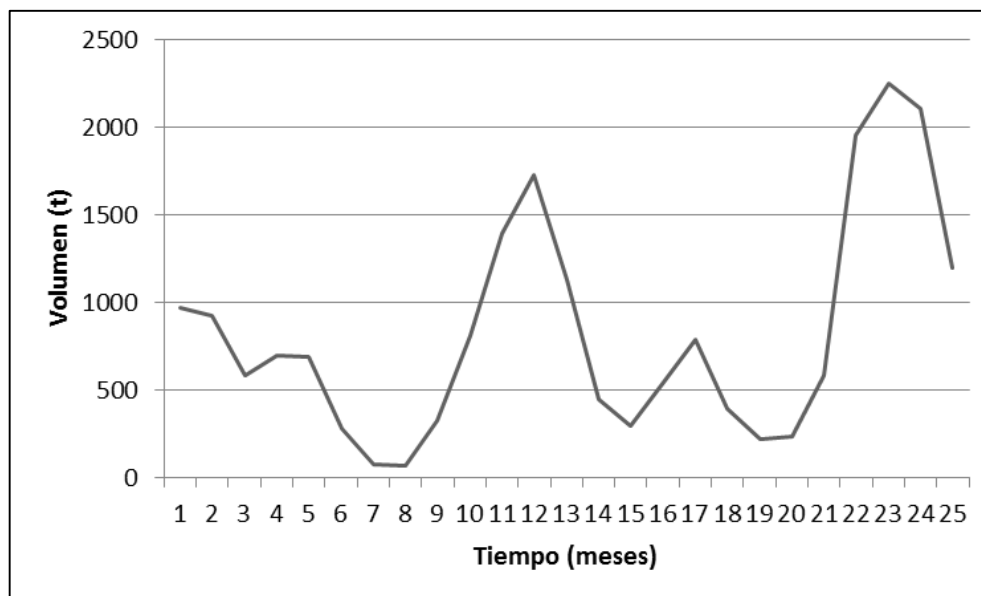


Figura 11. Serie de tiempo de los datos de validación.

Fuente: Elaboración propia.

- De enero 2010 a junio 2015: Datos para el uso de la red (ver tabla 4).

Tabla 4. Datos para el uso de la red.

Mes	Volumen de producción (t)					
	2010	2011	2012	2013	2014	2015
<i>Enero</i>	2045,56	1879,10	2141,03	1729,31	2106,11	2286,95
<i>Febrero</i>	1226,11	1750,52	965,62	1130,04	1194,87	756,85
<i>Marzo</i>	785,42	1618,00	925,8	447,29	506,10	623,23
<i>Abril</i>	368,85	1111,07	585,3	298,77	842,26	588,05
<i>Mayo</i>	343,49	1057,30	693,5	545,6	1070,08	514,35
<i>Junio</i>	340,95	1015,60	689,7	785,9	649,98	456,40
<i>Julio</i>	266,50	521,45	281,5	397,55	199,18	-
<i>Agosto</i>	644,52	115,20	78,75	219	151,13	-
<i>Septiembre</i>	720,95	105,80	67,4	233,35	203,33	-
<i>Octubre</i>	663,99	368,25	324,25	586,27	1191,80	-
<i>Noviembre</i>	1745,30	694,42	811,75	1953,4	2501,25	-
<i>Diciembre</i>	1484,55	1765,95	1390,65	2244,13	2526,75	-

Fuente: Hielos Santa Lucía S.R.L.

A continuación se grafica la serie de tiempo de los datos para la fase de entrenamiento (ver figura 12).

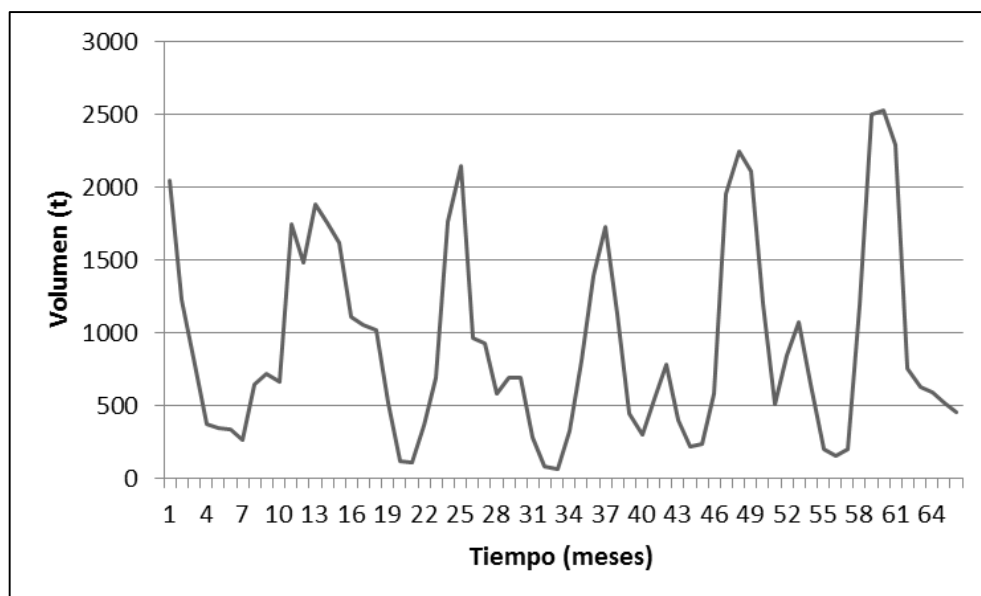


Figura 12. Serie de tiempo de los datos para el uso de la red.

Fuente: Elaboración propia.

### 3.1.3 Selección de la topología

Se analizó el comportamiento de las topologías teniendo como factor de variación el número de neuronas en la capa de entrada. La red fue de tipo perceptrón multicapa.

A partir de ensayo y error se ha visto que con 16 neuronas en la capa de entrada se obtienen óptimos resultados. A partir de estas 16 neuronas podremos predecir el siguiente valor de la serie.

De acuerdo a lo recomendado, el número de neuronas en la capa oculta debe ser menor que en la capa de entrada y equivalente a  $\frac{n*m}{2}$ , donde  $n$  es el número de neuronas en la capa de entrada y  $m$  el número de salidas que tiene la red, y considerando que en la salida de la red hay una neurona, da como resultado que el número de neuronas en la capa oculta es ocho (ver figura 13).

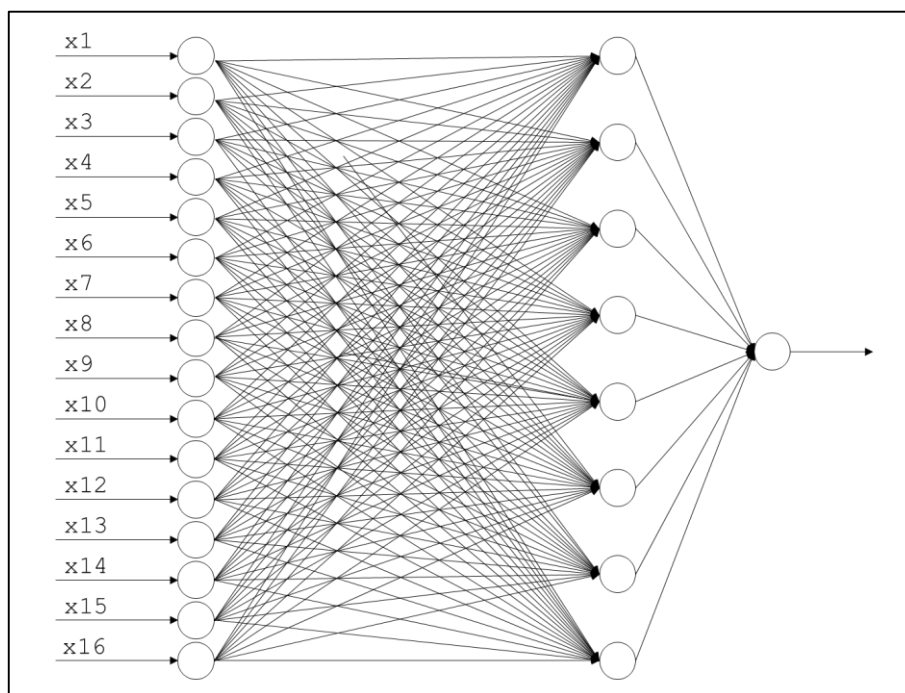


Figura 13. Esquema de una red multicapa.

Fuente: Elaboración propia.

### 3.1.4 Normalización de datos

El proceso de normalización de datos fue necesario para la investigación puesto que la respuesta de las neuronas viene dada por la función de activación sigmoide. Esta función está diseñada para devolver valores entre 0 y 1. La normalización de

los datos no supone ninguna pérdida de información. Con este proceso se obtiene la serie temporal multiplicada por un factor y desplazada de tal forma que los valores están dentro del rango  $[0,1]$ .

### **3.1.5 Fase de entrenamiento**

La fase de entrenamiento de una red neuronal sirvió para encontrar los pesos y umbrales óptimos, inicialmente aleatorios. En cada etapa se calculó el error cometido por la red cuando intenta predecir el resultado. La clave del proceso de entrenamiento fue modificar los pesos y umbrales en la dirección que minimiza el error. Para ello se usaron solo los patrones de entrenamiento y una reiteración de, a veces, cientos de etapas en las que se repiten los mismos pasos hasta llegar a la estabilización de los pesos y umbrales.

Al final del proceso, cuando los valores de los pesos y umbrales se estabilizaron asintóticamente, la red neuronal se dice que ha aprendido a pronosticar de forma aproximada la cantidad demandada de hielo industrial.

El proceso de entrenamiento de las redes neuronales en la herramienta de software matemático MATLAB se diseñó en varios pasos.

### **3.1.6 Implementación**

Se llevó a cabo la implementación del modelo haciendo uso de la técnica de Back Propagation para el perceptrón multicapa. Para proceder con la implementación de este modelo, antes presentamos un resumen de los datos (ver

tabla 5) más relevantes del mismo que nos ayudará a internarnos en el proceso de su implementación en la herramienta de software matemático MATLAB.

Tabla 5. Resumen de datos.

Data pre-procesada	66 registros
Tipo de variable	Cuantitativa (toneladas)
Frecuencia de data	Mensual
Número de épocas	250
Tamaño del conjunto de entrenamiento	25
Número de neuronas de entrada	16
Número de capas ocultas	1
Número de neuronas ocultas en la capa oculta	8
Número de neuronas de salida	1
Función de transferencia	Función sigmoide
Función de error	Error cuadrático medio

Fuente: Elaboración propia.

Se procedió a explicar las instrucciones necesarias para implementar el modelo, si desea estudiar el código completo (ver anexo 4).

Se comenzó a desarrollar el corazón matemático de las redes neuronales artificiales.

Las redes neuronales tipo perceptrón usaron el método del descenso del gradiente para minimizar la función error. Antes de explicarlo es conveniente ver porqué el método estándar de optimización del cálculo no es el adecuado. Supongamos que tenemos una función  $f(w)$ . El objetivo es calcular  $w^*$ , que es el valor de  $w$  que minimiza  $f(w)$ . Por ejemplo la función  $f(w) = (w - 1)^2$ . El valor que anula la derivada es el mínimo  $f'(w) = 2(w - 1) = 0 \rightarrow w^* = 1$ .

En redes neuronales este método no es efectivo porque la función a minimizar depende de demasiadas variables y tiene una forma demasiado compleja como para trabajar con ella de forma general.

El método del descenso del gradiente es el más usado. Veamos un ejemplo de cómo funciona a partir de un caso sencillo con una sola variable.

Tabla 6. Ejemplo de descenso del gradiente.

w	f(w)
2	1
2,1	1,21
1,9	0,81

Fuente: Elaboración propia.

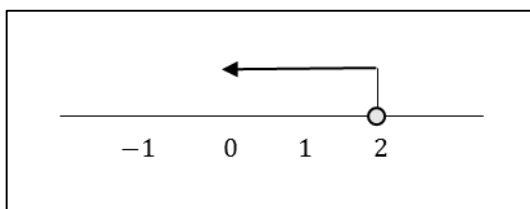


Figura 14. Demostración de rango.

Fuente: Elaboración propia.

Por inspección (ver tabla 6) si partimos del 2 porque la función es más pequeña (ver figura 14), la idea es como variar la  $w$  para acercarnos al mínimo. Lo que dice el descenso de gradiente es, que cuando tengamos que modificar la  $w$ , lo hagamos siempre en contra de la derivada  $\Delta w \sim -f'(w)$ . Si  $w_{(0)}$  entonces  $w_{(1)} = w_{(0)} - f'(w_{(0)})$ . Ejemplo:  $w_{(0)} = 2$ ;  $w_{(1)} = 2 - 2(2 - 1) = 0$ ;  $w_{(2)} = 0 - 2(0 - 1) = 2$ . Aplicando ésta regla, cuando queramos calcular la nueva  $w$  en nuestro camino

hacia el mínimo,  $\alpha$  como razón de aprendizaje y es un número muy pequeño (tiende a cero). Esta función nos lleva al mínimo más cercano  $w_{(n+1)} = w_{(n)} - \alpha f'(w_{(n)})$ .

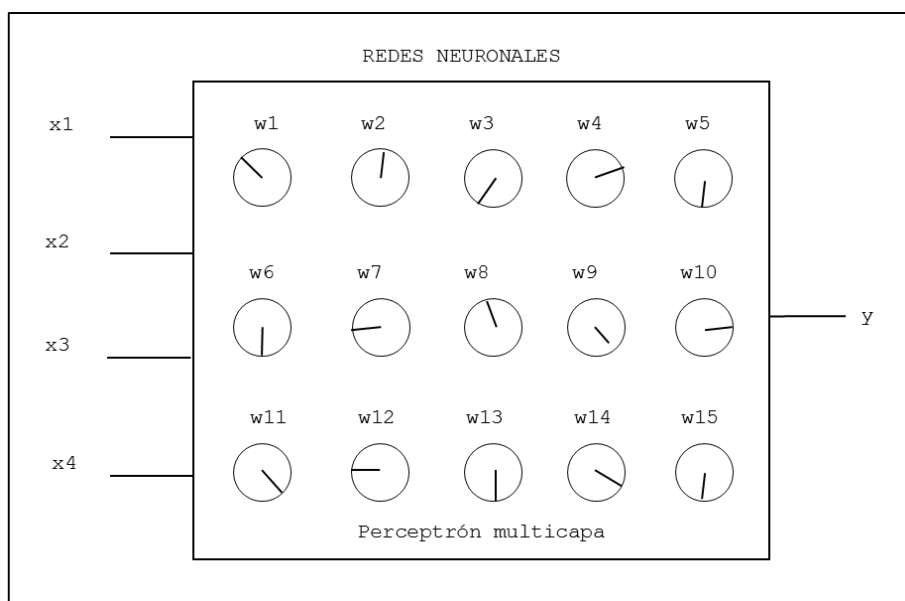


Figura 15. La red con pesos como voltímetros.

Fuente: Elaboración propia.

El error es una función y depende de las variables de las  $w$  (ver figura 15). Porque las entradas  $x$  y salidas  $y$  son valores fijos, porque las  $w$  son las que podemos modificar a nuestro gusto  $error(w_{(1)}, w_{(2)}, w_{(3)}, \dots)$ .

En el descenso del gradiente en un perceptrón, las variables son las  $w$ . Asumiendo que las  $w$  son independientes entre sí y también de cualquier otra variable de la red podremos usar derivadas parciales:  $w_i \rightarrow w_i - \alpha \frac{\partial error}{\partial w_i}$ .

A continuación vamos a derivar las ecuaciones del método Back Propagation. Empecemos con un caso sencillo. Consideremos la red neuronal como una caja negra con tres entradas  $x_1, x_2, x_3$  y dos salidas  $y_1, y_2$ . Con dos capas ocultas con cuatro neuronas cada una,  $n$  representa el número de neuronas de capa. Interconectamos todas las neuronas entre sí (como una sinapsis en biología). Cada conexión tiene asociada una  $w$ . Ejemplo (ver figura 15)  $n_1 = 3, n_2 = 4, n_3 = 4$  y  $n_4 = 2$  dando  $w_{ij}^{(k)}$ ;  $i = 1, \dots, n_k$ ;  $j = 1, \dots, n_{k+1}$  (ver tabla 7).

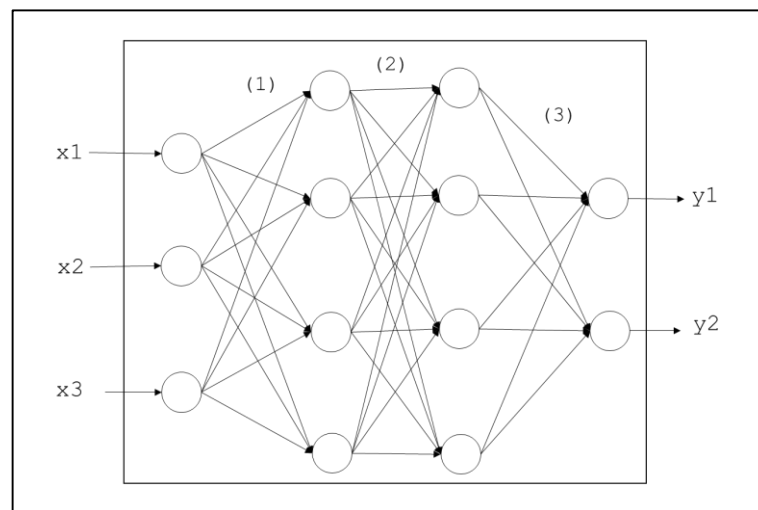


Figura 16. La red como una caja negra.

Fuente: Elaboración propia.

Tabla 7. Resultado de los pesos por capa.

(1)	(2)	(3)
$w_{ij}^{(1)}$	$w_{ij}^{(2)}$	$w_{ij}^{(3)}$
$i = 1, \dots, n_1$	$i = 1, \dots, n_2$	$i = 1, \dots, n_3$
$j = 1, \dots, n_2$	$j = 1, \dots, n_3$	$j = 1, \dots, n_4$

Fuente: Elaboración propia.

En el comportamiento de las neuronas (ver figura 16), supongamos que le llegan los números  $z_1, z_2, z_3$ , tendrá las conexiones que vienen desde otras neuronas  $w_1, w_2, w_3$  y tiene además como una especie de entrada que tiene un peso  $u$ . Se inspira en una red neuronal biológica como las que tenemos en nuestro cerebro pero es un modelo matemático simplificado, lo que ocurre es que todas las neuronas están conectadas a otras mediante las entradas, entran un número  $z$  según sea más fuerte hará que la señal que entre por la entrada sea más fuerte por eso a  $w$  se le llama peso, cuanto mayor sea este número mayor contribución tendrá o intensidad numérica. El umbral o umbral de activación  $u$ , si este es muy grande tanto positivo o negativo hace que la neurona se excite al máximo y dé una salida de uno o que suceda lo contrario y dé una de cero, ambos casos independientemente de los demás valores de entrada.

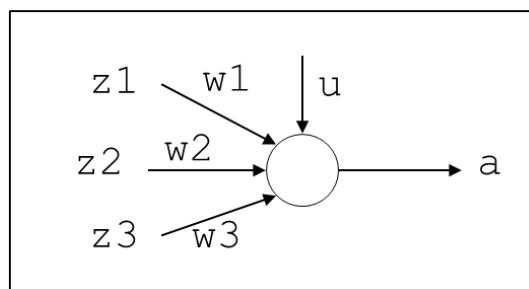


Figura 17. Comportamiento de una red.

Fuente: Elaboración propia.

Se puede simplificar con una función de activación, existen varias pero hay una que se utiliza mucho que se llama *sigmoide* (ver figura 17), si estamos muy a la derecha dará 1 y si estamos muy a la izquierda dará cero.

La función sigmoide es:  $f(x) = (1 + e^{-x})^{-1}$  la derivada  $f'(x) = +(1 + e^{-x})^{-2}e^{-x}$  se puede comprobar fácilmente que  $f'(x) = f(x)(1 - f(x))$  veamos, tenemos  $f = \frac{1}{1+e^{-x}} \rightarrow f^2 = \frac{1}{(1+e^{-x})^2}$  luego que  $1 + e^{-x} = \frac{1}{f} \rightarrow e^{-x} = \frac{1}{f} - 1$ , entonces  $f'(x) = f^2(x)e^{-x} = f^2\left(\frac{1}{f} - 1\right) \rightarrow f'(x) = f(x)(1 - f(x))$ . Luego veremos que esta función será muy útil.

Recordamos la regla de la cadena  $[f(\Theta)]' = f'(\Theta) \cdot \Theta'$ , en dónde  $\Theta$  representa una función cualquiera. Por ejemplo si tenemos  $f(x) = x^2$  y  $\Theta = g(x) = \sin x \rightarrow \Theta' = \cos x$ , si queremos derivar  $[f(\Theta)]' = 2\Theta \cdot \Theta'$ , entonces  $[f(\sin x)]' = 2 \sin x \cdot \cos x$ .

Si la función de transferencia  $f(x)$  es la sigmoide entonces se puede ver fácilmente que su derivada es proporcional a  $f'(x) = f(x)(1 - f(x))$ , entonces  $[f(\Theta)]' = f'(\Theta) \cdot \Theta' \rightarrow [f(\Theta)]' = f(\Theta)(1 - f(\Theta)) \cdot \Theta'$ .

La salida de la neurona  $a = f(u + w_1z_1 + w_2z_2 + w_3z_3)$ . Vamos a hallar la derivada parcial de  $a$  con respecto de  $u$ , para saber cómo va a cambiar  $a$  si lo único que va a cambiar es  $u$  y lo demás sigue constante, tenemos que  $a = f(\Theta)$  por tanto  $\frac{\partial a}{\partial u} = \frac{\partial f}{\partial u} \Big|_{\Theta} \cdot \Theta' = f(\Theta)(1 - f(\Theta)) = a(1 - a)$ . En el caso de  $w$  es parecido  $\frac{\partial a}{\partial w_1} = a(1 - a) \cdot \Theta' = a(1 - a)z_1$ .

La nomenclatura del gráfico, las salidas será en general  $a_i^{(k)}$ ,  $i = 1, \dots, n_k$ ,  $k$  específica en que capa estamos. Las neuronas de entrada son los valores de entrada, es decir cuando  $k = 1$  tenemos  $a_i^{(1)} = x_i$  y las demás serán  $a_i^{(k)} = f(\otimes)$ . Por ejemplo  $a_1^{(2)} = f(u_1^{(2)} + x_1 w_{11}^{(1)} + x_2 w_{21}^{(1)} + x_3 w_{31}^{(1)})$ , generalizando cuando  $k > 1$  tenemos  $a_i^{(2)} = f(u_i^{(2)} + a_1^{(1)} w_{1i}^{(1)} + a_2^{(1)} w_{2i}^{(1)} + a_3^{(1)} w_{3i}^{(1)})$  y es lo mismo que  $a_i^{(k)} = f(u_i^{(k)} + \sum_{j=1}^{n_{k-1}} a_j^{(k-1)} w_{ji}^{(k-1)})$  para  $i = 1, \dots, n_k$ . La salida cuando  $k = 4$  tenemos  $y_i \equiv a_i^{(4)}$ .

El Back Propagation es el algoritmo con que aprende la red neuronal.

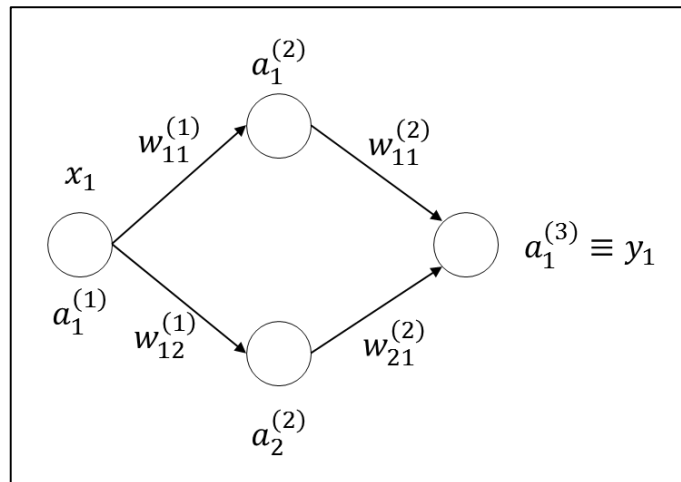


Figura 18. Ejemplo simple de arquitectura del Perceptrón.

Fuente: Elaboración propia.

La expresión de  $a_1^{(3)}$  (ver figura 18) será  $a_1^{(3)} = f(u_1^{(3)} + w_{11}^{(2)} a_1^{(2)} + w_{21}^{(2)} a_2^{(2)})$  donde  $a_1^{(2)} = f(u_1^{(2)} + w_{11}^{(1)} a_1^{(1)})$  y  $a_2^{(2)} = f(u_2^{(2)} + w_{12}^{(1)} a_1^{(1)})$ , quedando  $a_1^{(3)} =$

$f(u_1^{(3)} + w_{11}^{(2)}f(u_1^{(2)} + w_{11}^{(1)}a_1^{(1)}) + w_{21}^{(2)}f(u_2^{(2)} + w_{12}^{(1)}a_1^{(1)}))$ . Calcularemos las derivadas parciales de la salida con respecto de las variables  $\frac{\partial a_1^{(3)}}{\partial w_{11}^{(1)}} = a_1^{(3)}(1 - a_1^{(3)})w_{11}^{(2)} \cdot a_1^{(2)}(1 - a_1^{(2)})a_1^{(1)}$  y la estructura es la misma para  $\frac{\partial a_1^{(3)}}{\partial w_{12}^{(1)}} = a_1^{(3)}(1 - a_1^{(3)})w_{21}^{(2)} \cdot a_2^{(2)}(1 - a_2^{(2)})a_1^{(1)}$ , ambas en forma resumida será  $\frac{\partial a_1^{(3)}}{\partial w_{1j}^{(1)}} = a_1^{(3)}(1 - a_1^{(3)})w_{j1}^{(2)} \cdot a_j^{(2)}(1 - a_j^{(2)})a_1^{(1)}$  para  $j = 1, 2$ . También  $\frac{\partial a_1^{(3)}}{\partial w_{11}^{(2)}} = a_1^{(3)}(1 - a_1^{(3)})a_1^{(2)}$  por lo tanto tenemos  $\frac{\partial a_1^{(3)}}{\partial w_{j1}^{(2)}} = a_1^{(3)}(1 - a_1^{(3)})a_j^{(2)}$  para  $j = 1, 2$ . Ahora si queremos derivar con respecto al umbral  $\frac{\partial a_1^{(3)}}{\partial u_1^{(2)}} = a_1^{(3)}(1 - a_1^{(3)})w_{11}^{(2)}a_1^{(2)}(1 - a_1^{(2)})$ .

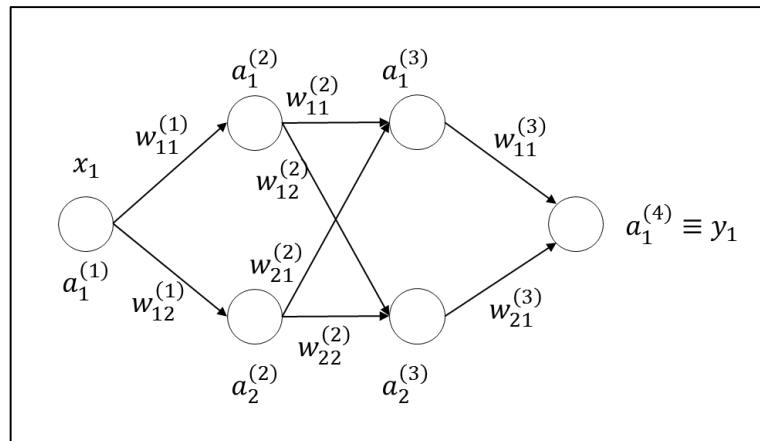


Figura 19. Ejemplo avanzado de arquitectura del Perceptrón.

Fuente: Elaboración propia.

A continuación se demostrará la regla en general, calculando gráficamente las derivadas (ver figura 19), comenzamos con  $\frac{\partial y_1}{\partial w_{11}^{(1)}} = a_1^{(4)}(1 - a_1^{(4)})w_{11}^{(2)}a_1^{(3)}(1 - a_1^{(3)})w_{11}^{(2)}a_1^{(2)}(1 - a_1^{(2)})a_1^{(1)} + a_1^{(4)}(1 - a_1^{(4)})w_{21}^{(3)}a_2^{(3)}(1 - a_2^{(3)})w_{12}^{(2)}a_1^{(2)}(1 - a_1^{(2)})a_1^{(1)}$  expresamos reordenando los términos de izquierda a derecha  $\frac{\partial y_1}{\partial w_{11}^{(1)}} = a_1^{(1)}a_1^{(2)}(1 - a_1^{(2)})w_{12}^{(2)}a_2^{(3)}(1 - a_2^{(3)})w_{21}^{(3)}a_1^{(4)}(1 - a_1^{(4)})$ , ahora generalizamos con sumatoria para indicar el número de caminos o bifurcaciones quedando  $\frac{\partial y_1}{\partial w_{11}^{(1)}} = x_1 a_1^{(2)}(1 - a_1^{(2)})[\sum_{j=1}^2 w_{1j}^{(2)} a_j^{(3)}(1 - a_j^{(3)})w_{j1}^{(3)}]y_1(1 - y_1)$ .

Ahora haremos lo más general posible  $i = 1, \dots, n_4, j = 1, \dots, n_1, k = 1, \dots, n_2$

comenzamos  $\frac{\partial y_i}{\partial w_{jk}^{(1)}} = x_j a_k^{(2)}(1 - a_k^{(2)})w_{kp}^{(2)}a_p^{(3)}(1 - a_p^{(3)})w_{pi}^{(3)}y_i(1 - y_i)$

quedando  $\frac{\partial y_i}{\partial w_{jk}^{(1)}} = x_j a_k^{(2)}(1 - a_k^{(2)})[\sum_{p=1}^{n_3} w_{kp}^{(2)}a_p^{(3)}(1 - a_p^{(3)})w_{pi}^{(3)}]y_i(1 - y_i)$  y

$$\frac{\partial y_i}{\partial w_{jk}^{(2)}} = a_j^{(2)}a_k^{(3)}(1 - a_k^{(3)})w_k^{(3)}y_i(1 - y_i).$$

En resumen expresamos con respecto a los pesos y umbrales lo encontrado:

$$\frac{\partial y_i}{\partial w_{ji}^{(3)}} = a_j^{(3)}y_i(1 - y_i), \frac{\partial y_i}{\partial u_i^{(3)}} = y_i(1 - y_i);$$

$$\frac{\partial y_i}{\partial w_{jk}^{(2)}} = a_j^{(2)}a_k^{(3)}(1 - a_k^{(3)})w_{ki}^{(3)}y_i(1 - y_i),$$

$$\frac{\partial y_i}{\partial u_j^{(2)}} = a_j^{(3)}(1 - a_j^{(3)})w_{ji}^{(3)}y_i(1 - y_i);$$

$$\frac{\partial y_i}{\partial w_{jk}^{(1)}} = x_j a_k^{(2)} (1 - a_k^{(2)}) \left[ \sum_{p=1}^{n_3} w_{kp}^{(2)} a_p^{(3)} (1 - a_p^{(3)}) w_{pi}^{(3)} \right] y_i (1 - y_i),$$

$$\frac{\partial y_i}{\partial u_j^{(1)}} = a_j^{(2)} (1 - a_j^{(2)}) \left[ \sum_{p=1}^{n_3} w_{jp}^{(2)} a_p^{(3)} (1 - a_p^{(3)}) w_{pi}^{(3)} \right] y_i (1 - y_i).$$

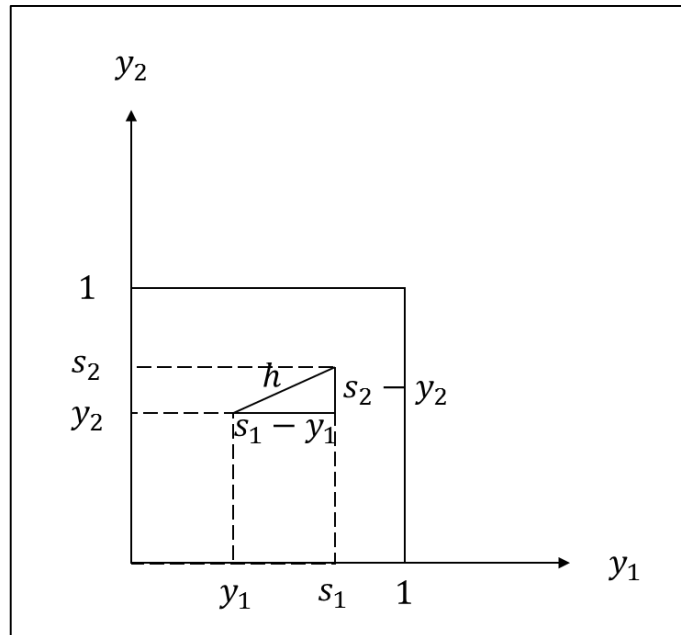


Figura 20. Ejemplo de aplicación de Pitágoras.

Fuente: Elaboración propia.

Veamos qué expresión matemática cuantifica la diferencia entre los valores calculados y los deseados. La manera estándar que resulta fructífera será la de la distancia geométrica Euclidiana en  $\mathbb{R}^n$ .

Según el gráfico (ver figura 20) tenemos las coordenadas  $(s_1, s_2)$  y  $(y_1, y_2)$ , se halla fácilmente  $h$  por la teoría de Pitágoras  $h = \sqrt{(s_1 - y_1)^2 + (s_2 - y_2)^2}$ . Se

infiere que podría ser una solución. Si tenemos una función  $f(x) = \sqrt{g(x)}$  si nos piden minimizar  $f'(x) = 0$  será  $f'(x) = \frac{g'(x)}{2\sqrt{g(x)}} = 0$  la única opción sería que  $g'(x) = 0$  en realidad estamos minimizando  $g(x) \geq 0$ .

Normalmente tenemos  $e = \frac{1}{2}(s_1 - y_1)^2 + \frac{1}{2}(s_2 - y_2)^2$ . Por ejemplo  $f(x, y)$ ,  $x(z_1, z_2, \dots)$ ,  $y(z_1, z_2, \dots)$  queremos encontrar la derivada parcial con respecto de variable en cuestión  $\frac{\partial f}{\partial z_1} = \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial z_1} + \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial z_1} \rightarrow \frac{\partial f}{\partial \square} = \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial \square} + \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial \square}$  aplicando esto a la función  $e$  derivando parcialmente  $\frac{\partial e}{\partial \square} = \frac{\partial e}{\partial y_1} \cdot \frac{\partial y_1}{\partial \square} + \frac{\partial e}{\partial y_2} \cdot \frac{\partial y_2}{\partial \square}$  se deduce  $\frac{\partial e}{\partial \square} = \sum_{i=1}^{n_4} -(s_i - y_i) \frac{\partial y_i}{\partial \square}$  entonces  $\frac{\partial e}{\partial y_i} = -(s_i - y_i)$ .

De acuerdo con lo solucionado anteriormente completamos los sumatorios.

$$\frac{\partial e}{\partial w_{ji}^{(3)}} = a_j^{(3)} y_i (1 - y_i) (-(s_i - y_i))$$

$$\frac{\partial e}{\partial w_{jk}^{(2)}} = a_j^{(2)} a_k^{(3)} (1 - a_k^{(3)}) \sum_{i=1}^{n_4} y_i (1 - y_i) [-(s_i - y_i)]$$

$$\frac{\partial e}{\partial w_{jk}^{(1)}} = x_j a_k^{(2)} (1 - a_k^{(2)}) \sum_{p=1}^{n_3} w_{kp}^{(2)} a_p^{(3)} (1 - a_p^{(3)}) \sum_{i=1}^{n_4} y_i (1 - y_i) [-(s_i - y_i)]$$

Observamos valores que son llamados nuevamente le asignamos valores  $\delta$ ,

como:  $y_i (1 - y_i) (-(s_i - y_i)) \equiv \delta_i^{(4)}$ ;  $a_k^{(3)} (1 - a_k^{(3)}) \sum_{i=1}^{n_4} y_i (1 - y_i) [-(s_i - y_i)] \equiv \delta_k^{(3)}$ ;

$$a_k^{(2)}(1 - a_k^{(2)}) \sum_{p=1}^{n_3} w_{kp}^{(2)} a_p^{(3)}(1 - a_p^{(3)}) \sum_{i=1}^{n_4} y_i(1 - y_i) [-(s_i - y_i)] \equiv \delta_k^{(2)}.$$

Simplificando los valores  $\delta$  asignados  $\frac{\partial e}{\partial w_{ji}^{(3)}} = a_j^{(3)} \delta_i^{(4)}$ ;  $\frac{\partial e}{\partial w_{jk}^{(2)}} = a_j^{(2)} \delta_k^{(3)}$ ;  $\frac{\partial e}{\partial w_{jk}^{(1)}} = x_j \delta_k^{(2)}$ ;  $\frac{\partial e}{\partial u_i^{(4)}} = \delta_i^{(4)}$ ;  $\frac{\partial e}{\partial u_i^{(3)}} = \delta_i^{(3)}$ ;  $\frac{\partial e}{\partial u_i^{(2)}} = \delta_i^{(2)}$ . Estas serían las fórmulas para las derivadas parciales con respecto de  $w$  y  $u$ . Recordando que  $\alpha$  es la razón de aprendizaje para  $w \rightarrow w - \alpha \frac{\partial e}{\partial w}$ ;  $u \rightarrow u - \alpha \frac{\partial e}{\partial u}$ . Sabemos que a la red solo podemos introducir valores entre cero y uno, porque trabajando con la función sigmoide, entonces aplicamos el escalamiento.

Tabla 8. Ejemplo de función de escalamiento.

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
5	4	3	3	-5
-1	2	8	1	4

Fuente: Elaboración propia.

Comenzamos calculando el mínimo y máximo de los valores, luego aplicar  $valor \rightarrow \frac{valor - valor_{mínimo}}{valor_{máximo} - valor_{mínimo}}$ . Por ejemplo (Ver tabla 8)  $5 \rightarrow \frac{5 - (-1)}{5 - (-1)} = 1$ ;  $-1 \rightarrow \frac{-1 - (-1)}{5 - (-1)} = 0$ ;  $2 \rightarrow \frac{2 - (-1)}{5 - (-1)} = 0,5$ . Comprobamos que el escalamiento transforma el -1 en 0 y el 5 en 1 (ver figura 21).

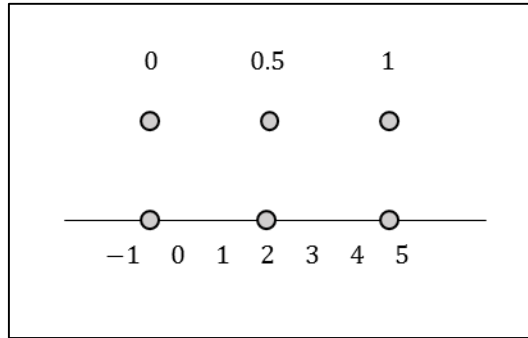


Figura 21. Ejemplo de escalamiento.

Fuente: Elaboración propia.

Para hallar la inversa de esta función  $\tilde{v} = \frac{v - \min}{\max - \min}$  despejando tenemos  $v = (\max - \min)\tilde{v} + \min$ .

Con esto llegamos a las fórmulas finales y ya podemos programar nuestro algoritmo, para ello comenzamos con la construcción de los script que es una secuencia de instrucciones de la herramienta de software matemático MATLAB guardada en un archivo con extensión *.m*, para la entrada de serie temporal preparamos los datos con *preparacion\_serie\_temporal.m* procedemos a leer *Normalizado.xlsx* que es un documento de Microsoft Excel con la serie normalizada, donde la variable *serie* guardamos los valores, construimos la estructura de *Normalizado\_patrones.xlsx* que es un nuevo documento de Microsoft Excel con el que aprenderá la red, almacena en *N* el número de datos de la serie, *longitud\_ventana* es el número par de valores del pasado para la predicción es decir, se introduce dieciséis datos de la serie para predecir el décimo séptimo dato, almacena en el arreglo *patron* y genera el arreglo *patron\_fila*, procedemos a escribir

en el documento el número de capas (3), el número de neuronas en la capa de entrada (16), el número de neuronas en la capa oculta (8), el número de neuronas en la capa salida (1), el número de patrones (50) y los patrones para la red.

Una vez finalizado *Normalizado\_patrones.xlsx*, empezamos con la construcción de *programa1.m* donde contiene funciones que pueden recibir varios valores y devolver varios resultados, se crea la función *excel.m* que devuelva los valores de entrada (*INPUT*), de salida (*OUTPUT*), el número total de capas (*capas*), el número de neuronas de cada capa (*n*) y el número de registros o patrones para el aprendizaje (*N*).

Se crea la función *escalamiento.m* que tiene como finalidad transformar los datos con normalización sigmoideal, recibe las entradas (*INPUT*) y salidas (*OUTPUT*) de los cuales calcula los mínimos (*INPUT\_min*, *OUTPUT\_min*) y máximos (*INPUT\_max*, *OUTPUT\_max*) de los valores de entrada y salida, iniciamos matrices para las entradas (*input*) y otra para salidas (*output*) donde se almacenará valores normalizados.

Según el proceso de aprendizaje del perceptrón multicapa – Paso 1 (ver anexo 3). Se crea la función *aleatorio.m* que tiene como finalidad crear pesos y umbrales aleatorios. Se crea matrices celulares multidimensionales *w* y *u* que sean de máxima entropía, cercano a cero. Se recorre capa a capa, en el primer condicional la capa de salida no tiene peso, con la distribución normal de números aleatorios devuelve una matriz de  $n(c)$ ,  $n(c+1)$  y al ser multiplicado por 0.1 se obtiene máxima entropía y

en el segundo condicional la capa de inicio no tiene umbral, también se trabaja con la distribución normal de números aleatorios devuelve una matriz de  $1, n(c)$ .

Los valores para la velocidad de aprendizaje ( $0,02\%$ ) y para el momento ( $0,99\%$ ) son adecuados para buscar la solución óptima. En el aprendizaje utilizamos los primeros  $N/2$  datos. Se crea matrices celulares multidimensionales *cambio\_peso* y *cambio\_umbral*. Momento iniciado a cero.

Se crea la función *reset\_momento.m* para el momento iniciado a cero, se recorre capa a capa, en el primer condicional la capa de salida no tiene peso, devuelve una matriz de ceros de  $n(c), n(c+1)$  y en el segundo condicional la capa de inicio no tiene umbral, devuelve una matriz de ceros de  $1, n(c)$ .

El número de etapa ( $250$ ) es de acuerdo al gráfico, visualizar si se empieza a estabilizar. Las variables *s* son las salidas esperadas que serán actualizadas.

Según el proceso de aprendizaje del perceptrón multicapa – Paso 2 (ver anexo 3) se crea la función *salida\_red.m* que calcula la salida de la red a partir de la función de activación *logsig* obteniendo las *a* de cada capa, excepto que en última capa son *y* las salidas de las neuronas.

Según el proceso de aprendizaje del perceptrón multicapa – Paso 4 (ver anexo 3) se crea la función *actualizacion.m* que contiene las deltas, el primer bucle hace un recorrido hacia atrás desde la penúltima hasta la segunda, es decir las capas medias, no extremos actualizando las *d*. El segundo bucle hace un recorrido desde

la primera capa hasta la penúltima actualizando los  $w$ . El tercer bucle hace un recorrido desde la segunda capa hasta la última actualizando los  $u$ .

Según el proceso de aprendizaje del perceptrón multicapa – Paso 3 (ver anexo 3) para visualizar el gráfico de evolución de error cuadrático utilizamos *plot* con los últimos  $N/2$  datos o resto de valores.

Según el proceso de aprendizaje del perceptrón multicapa – Paso 6 (ver anexo 3) se crea la función *valida.m* que devuelve el porcentaje de éxito, validando con el resto de valores, teniendo en cuenta los valores de salida y obtenidos por la red y los valores deseados  $s$ . En el condicional se aplica la fórmula de error, asumiendo que lo aceptable es un 0,5%.

Finalmente guardamos los pesos y umbrales óptimos, los valores máximos y mínimos de las entradas y salidas en el archivo *optimo1*.

Se crea la función *uso1.m* que tiene como finalidad utilizar la red neuronal. Para ello procedemos a leer *Normalizado.xlsx* que es un documento de Microsoft Excel con la serie normalizada, donde la variable *serie* guarda esos valores,  $N$  es el número de datos de la serie, *longitud\_ventana* es el número par de valores del pasado para la predicción es decir, se introduce dieciséis datos de la serie para predecir el décimo séptimo dato. Importamos los pesos y umbrales óptimos, los valores máximos y mínimos de las entradas y salidas del archivo *optimo1.m* que anteriormente han sido guardados. Se invoca a la función *salida\_red.m* que calcula la salida de la red a

partir de la función de activación *logsig* obteniendo las  $y$  de cada capa, excepto que en última capa son  $y$  las salidas de las neuronas. En  $s$  se almacena la última columna que son las salidas esperadas o deseadas de la red neuronal. Teniendo ya estructurado lo que debemos utilizar, pasamos a encontrar el error relativo.

## CAPÍTULO IV

### ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Al desarrollar el modelo de redes neuronales no existió previamente una estructura preliminar para poder resolver el problema que se plantea. Solo es afirmable de la existencia de una estructura, la cual pueda dar respuestas adecuadas al problema.

#### 4.1 Presentación de los resultados obtenidos en la investigación

Los datos pertenecientes a una distribución normal se pueden estandarizar o normalizar, lo cual se consigue utilizando la fórmula (ver anexo 3 - fórmula 1.23). Se procedió a convertir cada uno de los datos nominales resultando, los datos de entrenamiento (ver tabla 9), de validación (ver tabla 10) y de uso (ver tabla 11).

Tabla 9. Datos de entrenamiento.

	Mes	Volumen de producción (t)		
		2010	2011	2012
Datos para la fase de entrenamiento	<i>Enero</i>	0,804342611	0,73665806	0,843161811
	<i>Febrero</i>	0,471144815	0,684375953	-
	<i>Marzo</i>	0,291955191	0,630491797	-
	<i>Abril</i>	0,122573038	0,424368227	-
	<i>Mayo</i>	0,11226137	0,402504727	-
	<i>Junio</i>	0,111228577	0,385549027	-
	<i>Julio</i>	0,08095635	0,184621953	-
	<i>Agosto</i>	0,234663631	0,01943603	-
	<i>Septiembre</i>	0,265740948	0,015613882	-
	<i>Octubre</i>	0,242580357	0,122329071	-
	<i>Noviembre</i>	0,682253441	0,254953545	-
	<i>Diciembre</i>	0,576229492	0,690649968	-

Fuente: Elaboración propia.

Tabla 10. Datos de validación.

	Mes	Volumen de producción (t)		
		2012	2013	2014
Datos para la fase de validación	<i>Enero</i>	-	0,675751723	0,828962937
	<i>Febrero</i>	0,365226584	0,432081648	0,458440238
	<i>Marzo</i>	0,349035314	0,154467644	-
	<i>Abril</i>	0,210584097	0,094077703	-
	<i>Mayo</i>	0,254579462	0,194441621	-
	<i>Junio</i>	0,253034338	0,292150365	-
	<i>Julio</i>	0,087055523	0,134242788	-
	<i>Agosto</i>	0,004615041	0,061642304	-
	<i>Septiembre</i>	0	0,067477179	-
	<i>Octubre</i>	0,104438165	0,210978511	-
	<i>Noviembre</i>	0,302661272	0,766869295	-
	<i>Diciembre</i>	0,538048671	0,885083457	-

Fuente: Elaboración propia.

Tabla 11. Datos para el uso de la red.

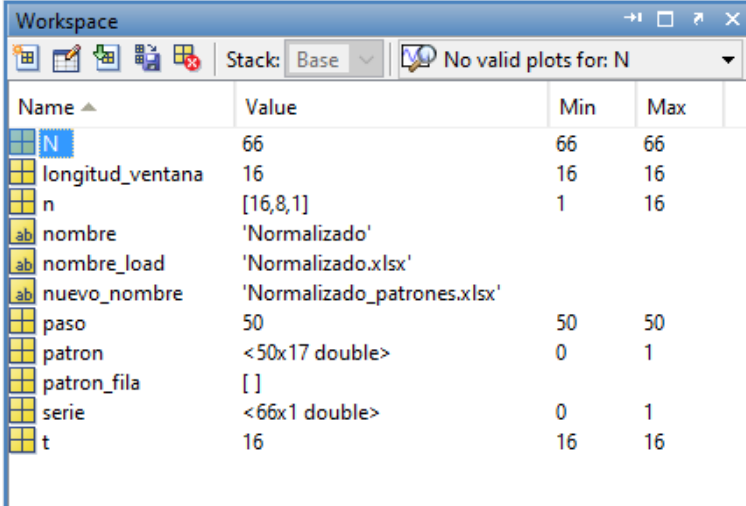
	Mes	Volumen de producción (t)					
		2010	2011	2012	2013	2014	2015
Datos para el uso de la red	<i>Enero</i>	0,804342611	0,73665806	0,843161811	0,675751723	0,828962937	0,9024946
	<i>Febrero</i>	0,471144815	0,684375953	0,365226584	0,432081648	0,458440238	0,2803383
	<i>Marzo</i>	0,291955191	0,630491797	0,349035314	0,154467644	0,1783784	0,2260048
	<i>Abril</i>	0,122573038	0,424368227	0,210584097	0,094077703	0,315065	0,2117023
	<i>Mayo</i>	0,11226137	0,402504727	0,254579462	0,194441621	0,4076992	0,181735
	<i>Junio</i>	0,111228577	0,385549027	0,253034338	0,292150365	0,2368817	0,1581719
	<i>Julio</i>	0,08095635	0,184621953	0,087055523	0,134242788	0,0535812	-
	<i>Agosto</i>	0,234663631	0,01943603	0,004615041	0,061642304	0,0340435	-
	<i>Septiembre</i>	0,265740948	0,015613882	0	0,067477179	0,0552687	-
	<i>Octubre</i>	0,242580357	0,122329071	0,104438165	0,210978511	0,457194	-
	<i>Noviembre</i>	0,682253441	0,254953545	0,302661272	0,766869295	0,9896314	-
	<i>Diciembre</i>	0,576229492	0,690649968	0,538048671	0,885083457	1	-

Fuente: Hielos Santa Lucía S.R.L.

## 4.2 Análisis e interpretación de los resultados obtenidos

### 4.2.1 Ejecución de `preparacion_serie_temporal.m`

Una vez ejecutado `preparacion_serie_temporal.m` se tuvo los siguientes resultados de las variables:  $N$ , `longitud_ventana`,  $n$ , `nombre`, `nombre_load`, `nuevo_nombres`, `paso`, `patron`, `patrón_fila`, `serie` y  $t$ . A continuación describiremos de forma global (ver figura 22), para ver información detallada ver anexo 5.



Name	Value	Min	Max
N	66	66	66
longitud_ventana	16	16	16
n	[16,8,1]	1	16
nombre	'Normalizado'		
nombre_load	'Normalizado.xlsx'		
nuevo_nombre	'Normalizado_patrones.xlsx'		
paso	50	50	50
patron	<50x17 double>	0	1
patron_fila	[]		
serie	<66x1 double>	0	1
t	16	16	16

Figura 22. Variables y valores de `preparacion_serie_temporal.m`.

Fuente: Elaboración propia.

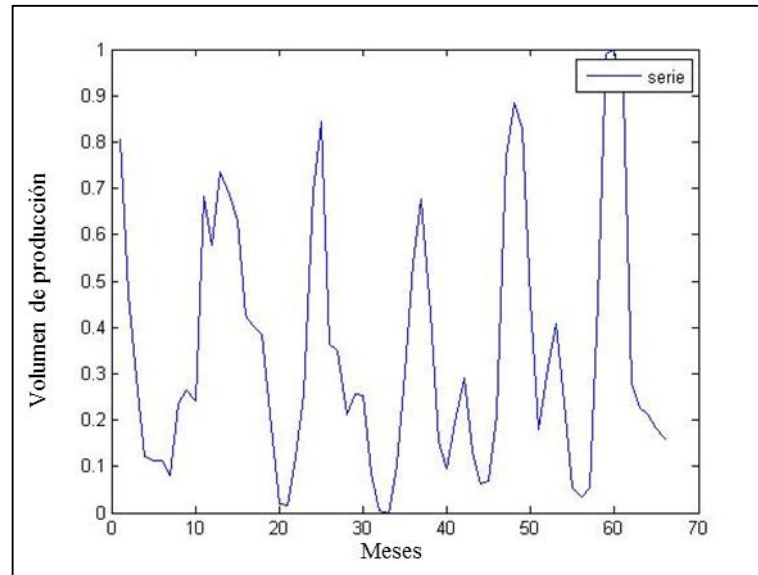


Figura 23. Gráfico de líneas 2D utilizando ejes lineales de *serie*.

Fuente: Elaboración propia.

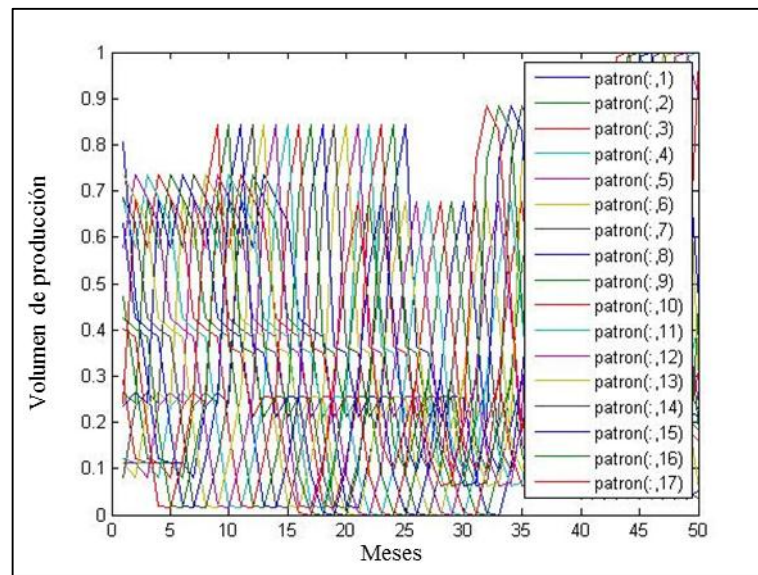


Figura 24. Gráfico de líneas 2D utilizando ejes lineales de *patron*.

Fuente: Elaboración propia.

Se trabajó con series de tiempo, porque a menudo puede observarse que una variable numérica alcanza valores por debajo del valor mínimo presenciado hasta el momento, o por encima del máximo. Se experimentó con la serie de demanda de hielo industrial (ver figura 23).

El aprendizaje de diferenciar patrones mediante ejemplos de entrenamientos, no es necesario elaborar modelos *a priori* ni necesidad de especificar funciones de distribución de probabilidad. Creará su propia representación característica, mediante la cual puede reconocer ciertos patrones (ver figura 24).

#### **4.2.2 Ejecución de programa1.m**

Una vez ejecutado *programa1.m* tenemos los siguientes resultados de las variables: *INPUT*, *INPUT\_max*, *INPUT\_min*, *N*, *OUTPUT*, *OUTPUT\_max*, *OUTPUT\_min*, *a*, *alfa*, *cambio\_peso*, *cambio\_umbral*, *capas*, *e*, *e\_media*, *eta*, *etapa*, *input*, *n*, *nombre*, *output*, *p*, *s*, *u*, *validacion*, *w*, *x* e *y*. A continuación describiremos de forma global (ver figura 25), se detalló información en el anexo 5.

Name	Value	Min	Max
INPUT	<50x16 double>	0	1
INPUT_max	<1x16 double>	0.8851	1
INPUT_min	<1x16 double>	0	0
N	50	50	50
OUTPUT	<50x1 double>	0	1
OUTPUT_max	1	1	1
OUTPUT_min	0	0	0
a	<1x3 cell>		
alfa	0.0200	0.0200	0.0200
ans	[-1.2335e-05;-7.4340e-05;1.412...]	-1.823...	2.2083...
cambio_peso	<1x2 cell>		
cambio_umbral	<1x3 cell>		
capas	3	3	3
e	<250x25 double>	4.3043...	0.2295
e_media	<1x250 double>	5.0997...	0.0313
eta	0.9900	0.9900	0.9900
etapa	250	250	250
input	<50x16 double>	0	1
n	[16,8,1]	1	16
nombre	'Normalizado_patrones.xlsx'		
output	<50x1 double>	0	1
p	25	25	25
s	0.1944	0.1944	0.1944
u	<1x3 cell>		
validacion	72	72	72
w	<1x2 cell>		
x	<1x16 double>	0	0.9526
y	0.2305	0.2305	0.2305

Figura 25. Variables y valores de *programa1.m*.

Fuente: Elaboración propia.

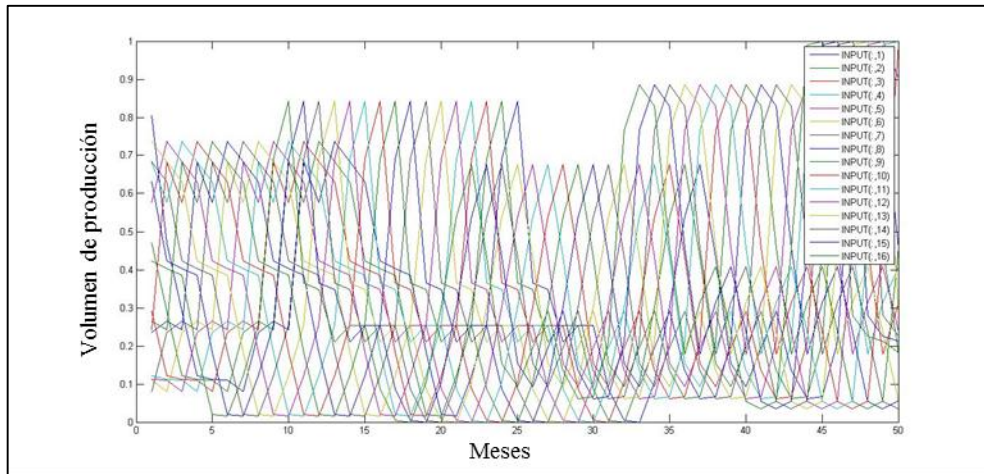


Figura 26. Gráfico de líneas 2D utilizando ejes lineales de *INPUT*.

Fuente: Elaboración propia.

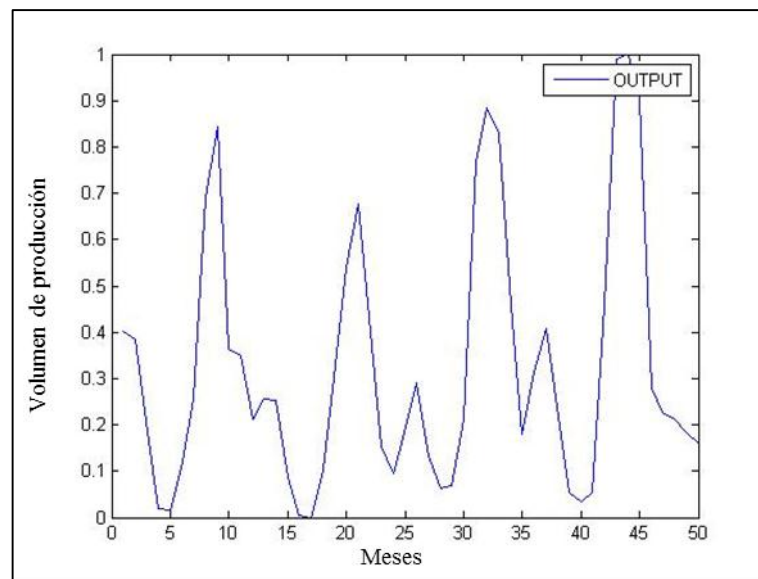


Figura 27. Gráfico de líneas 2D utilizando ejes lineales de *OUTPUT*.

Fuente: Elaboración propia.

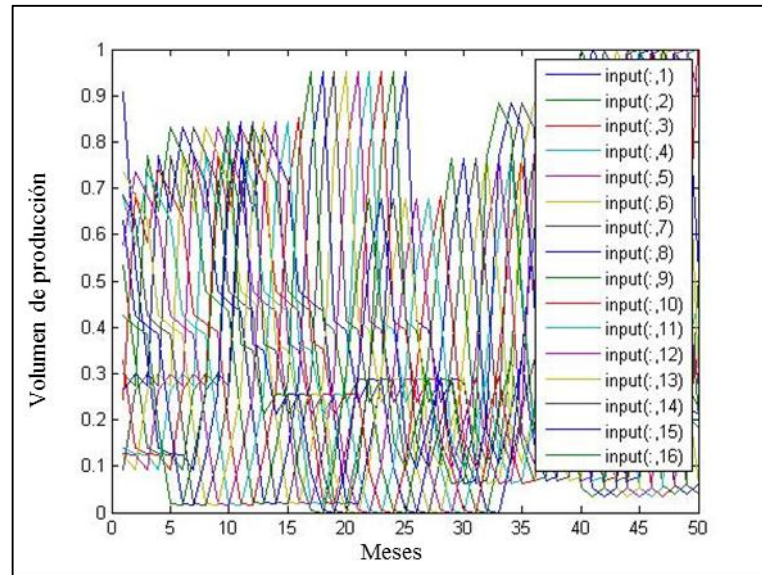


Figura 28. Gráfico de líneas 2D utilizando ejes lineales de *input*.

Fuente: Elaboración propia.

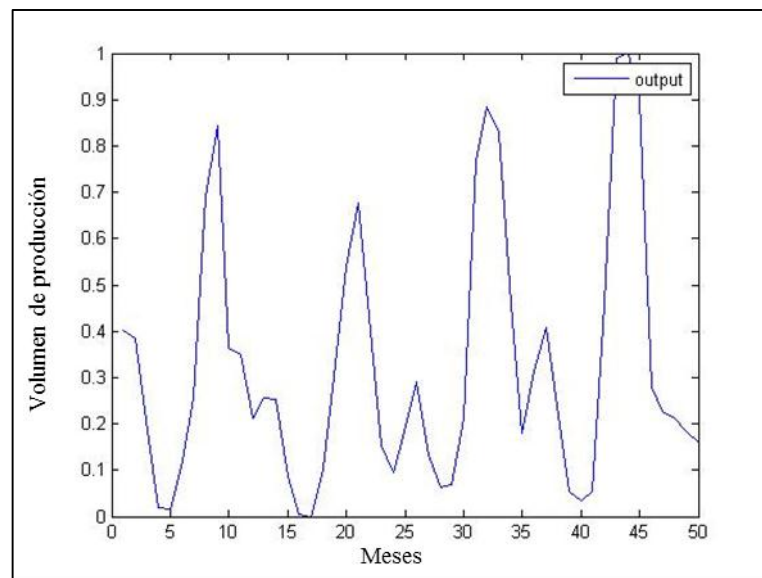


Figura 29. Gráfico de líneas 2D utilizando ejes lineales de *output*.

Fuente: Elaboración propia.

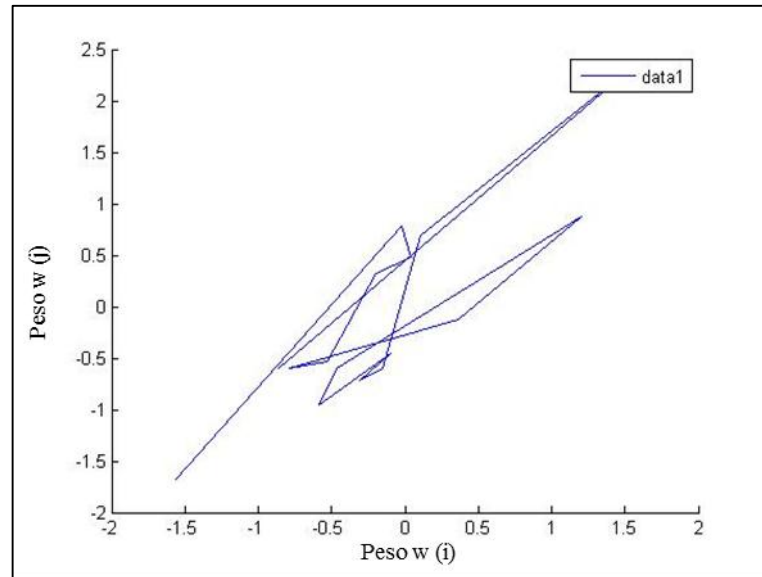


Figura 30. Gráfico de líneas 2D utilizando ejes lineales de  $w$ .

Fuente: Elaboración propia.

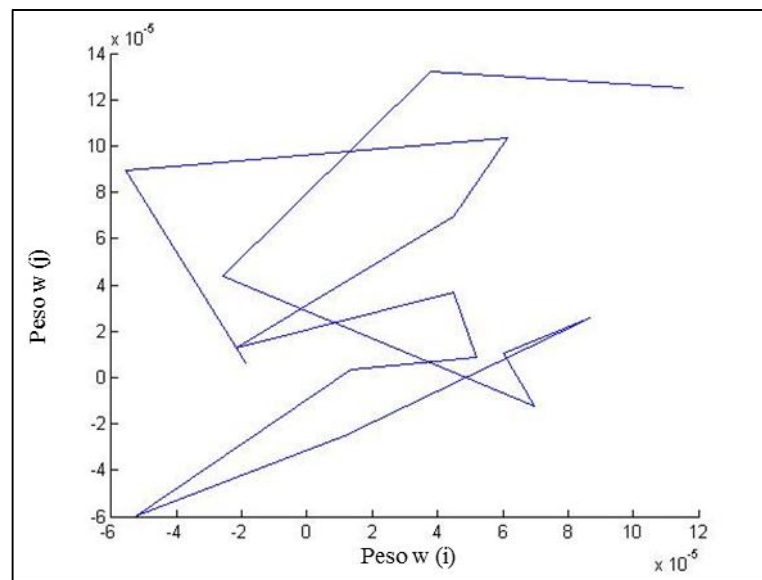


Figura 31. Gráfico de líneas 2D utilizando ejes lineales de  $cambio\_peso$ .

Fuente: Elaboración propia.

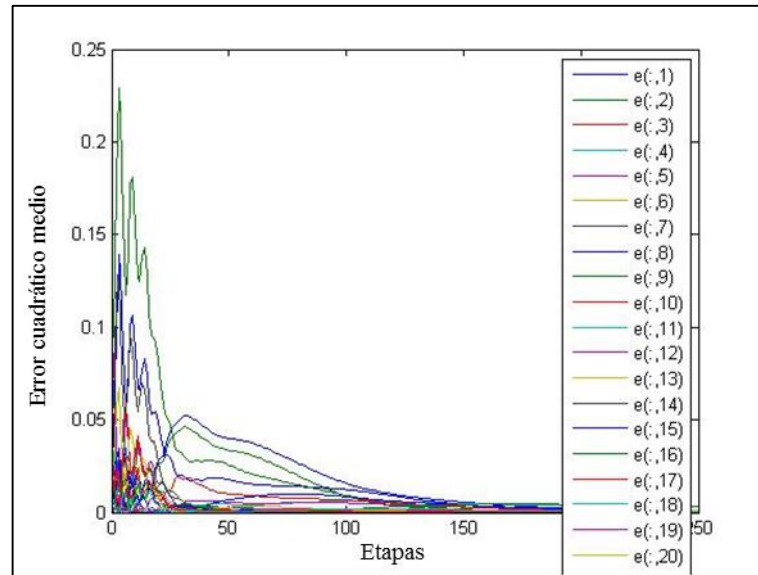


Figura 32. Gráfico de líneas 2D utilizando ejes lineales de  $e$ .

Fuente: Elaboración propia.

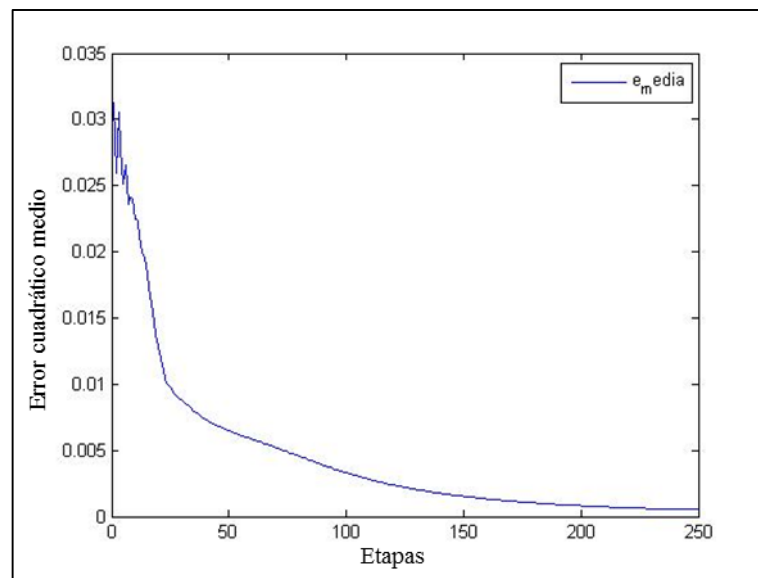


Figura 33. Gráfico de líneas 2D utilizando ejes lineales de  $e\_media$ .

Fuente: Elaboración propia.

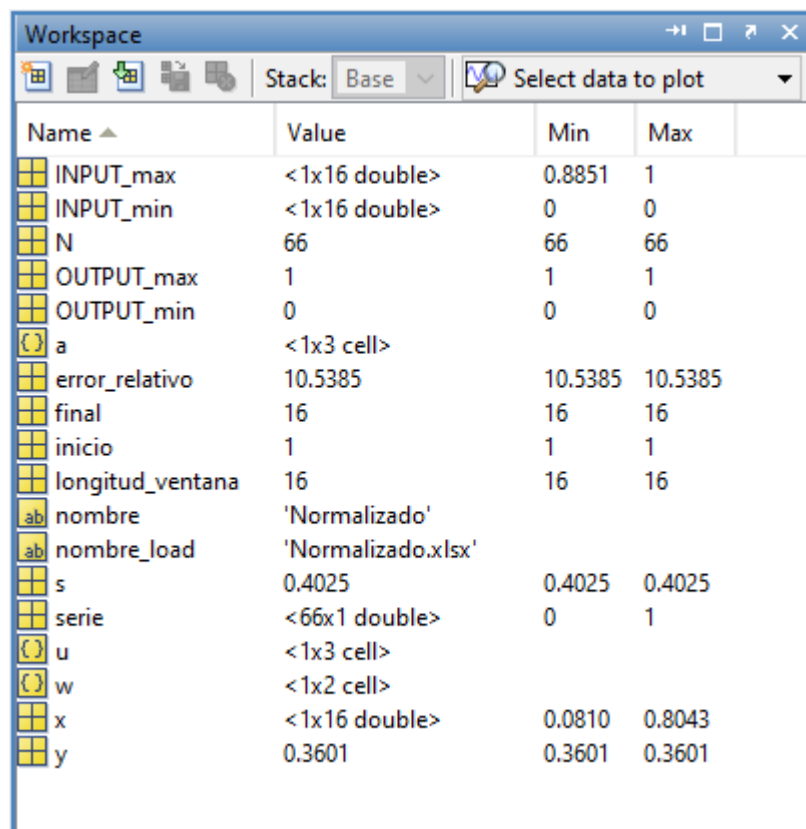
La estructura de entrada y salida fue definida mediante la hoja de cálculo Excel ingresado inicialmente *Normalizado.xlsx* que almacenó los valores de entrenamiento y validación. Se obtuvo los valores de los patrones sin normalizar de entrada (ver figura 26) y de salida (ver figura 27). Una vez definido se procedió a escalar con los máximos y mínimos de cada columna para la entrada (ver figura 28) y la salida (ver figura 29).

Los pesos son coeficientes que pueden adaptarse dentro de la red que determinan la intensidad de la señal de entrada registrada por la neurona (ver figura 30). Ellos son la medida de la fuerza de una conexión de entrada. Estas fuerzas pueden ser modificadas en respuesta de los ejemplos de entrenamiento (ver figura 31).

El error de entrenamiento se estabilizó, precisamente, cuando las salidas de la red no cambiaron de una iteración a otra, porque los parámetros de la red sufrieron cambios insignificantes de una iteración a otra (ver figura 32). La red evolucionó hasta estabilizarse ofreciendo exactitud en la solución que logra luego de varias iteraciones de la red neuronal, es decir la evolución del error a lo largo del proceso de aprendizaje (ver figura 33). Como resultado final el valor de test de validación fue de 72%.

### 4.2.3 Ejecución de *uso1.m*

Una vez ejecutado *uso1.m* tuvimos los siguientes resultados de las variables: *INPUT\_max*, *INPUT\_min*, *N*, *OUTPUT\_max*, *OUTPUT\_min*, *a*, *error\_relativo*, *final*, *inicio*, *longitud\_ventana*, *nombre*, *nombre\_load*, *s*, *serie*, *u*, *w*, *x* e *y*. A continuación describiremos de forma global (ver figura 34), se detalló información en el anexo 5.



Name ▲	Value	Min	Max
INPUT_max	<1x16 double>	0.8851	1
INPUT_min	<1x16 double>	0	0
N	66	66	66
OUTPUT_max	1	1	1
OUTPUT_min	0	0	0
a	<1x3 cell>		
error_relativo	10.5385	10.5385	10.5385
final	16	16	16
inicio	1	1	1
longitud_ventana	16	16	16
nombre	'Normalizado'		
nombre_load	'Normalizado.xlsx'		
s	0.4025	0.4025	0.4025
serie	<66x1 double>	0	1
u	<1x3 cell>		
w	<1x2 cell>		
x	<1x16 double>	0.0810	0.8043
y	0.3601	0.3601	0.3601

Figura 34. Variables y valores de *uso1.m*.

Fuente: Elaboración propia.

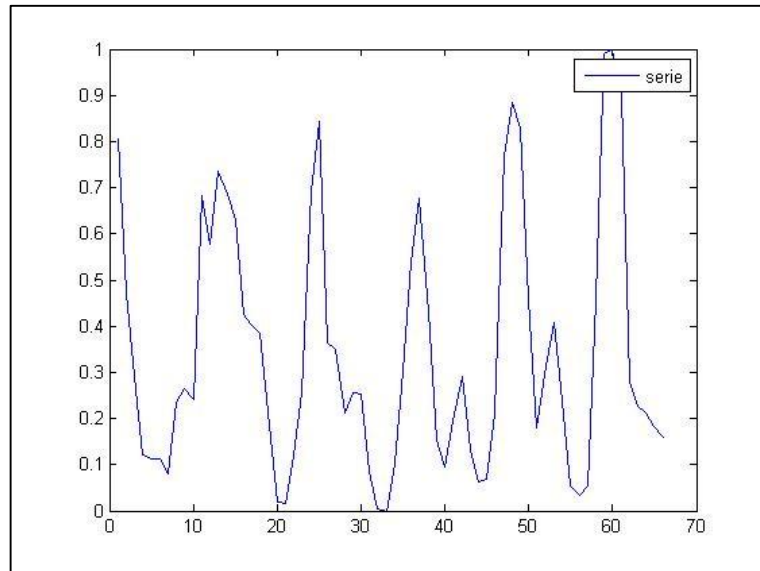


Figura 35. Gráfico de líneas 2D utilizando ejes lineales de *serie*.

Fuente: Elaboración propia.

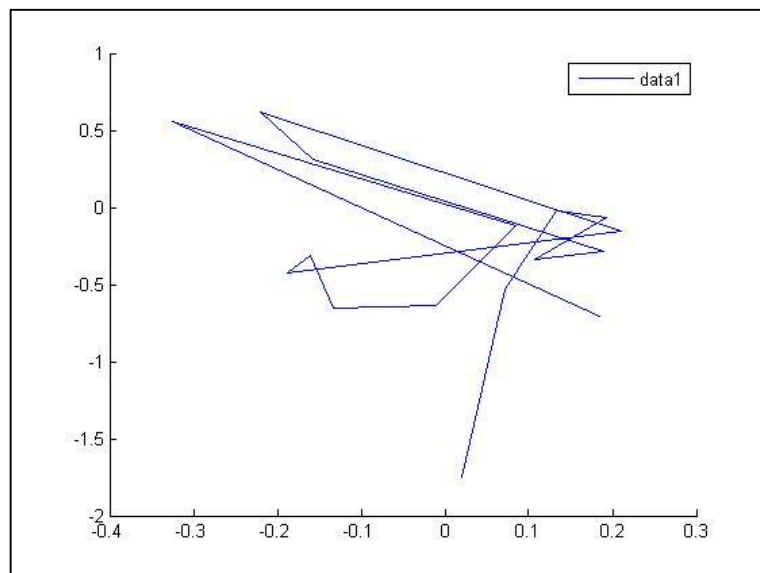


Figura 36. Gráfico de líneas 2D utilizando ejes lineales de  $w$ .

Fuente: Elaboración propia.

El uso de la red neuronal artificial tuvo como entrada la serie temporal (ver figura 35), y del resultado anterior como los pesos (ver figura 36), umbrales y el escalamiento fueron importados por *optimo1*. Se calculó el error de la red entre el valor real y el valor de predicción de la red neuronal artificial. Como resultado final el valor del error cuadrático medio es de 10%.

#### 4.2.4 Predicción

El pronóstico de demanda de hielo industrial en Hielos Santa Lucía S.R.L. utilizando la red neuronal artificial para los meses desde julio a diciembre 2015, (ver tabla 12) con resultado final, el valor de test de validación fue de 72% y el valor del error cuadrático medio fue de 10%.

Tabla 12. Predicción de demanda.

Año	Mes	(t)
2015	Julio	74,78
	Agosto	67,94
	Septiembre	148,56
	Octubre	1336,18
	Noviembre	2274,42
	Diciembre	2293,60

Fuente: Elaboración propia.

#### 4.2.5 Discusiones

En el transcurso de la investigación fue muy importante la recopilación de información, en búsqueda de una visión panorámica de los principios, desarrollo y aplicaciones de una red neuronal artificial enfatizando la capacidad para resolver problemas de predicción de series de tiempo.

Gómez (2010) concluye que las redes neuronales artificiales proporcionan buenas predicciones comparadas con las predicciones del modelo de serie de tiempo salvo que se tuvo que buscar la estructura idónea 1:2:16:12:1. Según el resultado de la presente tesis se obtuvo una estructura de 16:8:1; se ratifica que la topología es distinta de acuerdo al tipo de problema. Sin embargo, se difiere el resultado, el número de neuronas de la capa oculta debe ser menor a las de la capa de entrada.

Jiménez (2013) hizo una comparación y evaluación con el método de descomposición, Winter y ARIMA de sus resultados determinando que las redes neuronales artificiales son un modelo que obtiene mejores resultados de predicción con un error cuadrático medio del 6,22%. Según el resultado de la presente tesis se obtuvo un error cuadrático medio del 10%; se acepta la determinación.

Finalmente, Yanarico (2013) obtuvo resultados con una validación del 95%. Según el resultado de la presente tesis se obtuvo una validación del 72%; se discrepa con el investigador por ser un valor excesivo para una predicción.

## CONCLUSIONES

**PRIMERA:** Así mismo, al profundizar los conocimientos y al desarrollarlos, facilitó el diseño del modelo de aprendizaje para el funcionamiento de la red neuronal artificial con una topología de 16:8:1, se emplearon conceptos aprendidos durante la carrera, como: inteligencia artificial, programación matemática, álgebra lineal, simulación de sistemas, entre otros. Al tener un enfoque con sólida fundamentación matemática.

**SEGUNDA:** Se calculó los valores futuros a partir del modelo de predicción de demanda, fueron muy buenos, con el test de validación del 72% y el error cuadrático medio del 10%. Una de las características principales que se observó en el uso de la red neuronal artificial fue que funciona como una caja negra, es decir la interacción que se hizo con la red se centró en el inicio y en el final.

**TERCERA:** La investigación ha demostrado que se puede realizar la predicción de la demanda de hielo industrial en Hielos Santa Lucía S.R.L. utilizando redes neuronales artificiales, se pudo predecir los valores futuros de julio a diciembre del 2015. Los resultados de la investigación apoyan la utilización de las redes neuronales artificiales como técnica confiable en la predicción de series.

## RECOMENDACIONES

**PRIMERA:** Profundizar la investigación para definir el conjunto de datos razonables, dotar a la red neuronal artificial de suficientes parámetros para que sea capaz de aprender y evitar el sobreaprendizaje, que es el principal aspecto a tener en cuenta, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos. Los modelos de tipo caja negra pueden producir buenos resultados durante el aprendizaje y validación. Sin embargo, si el fenómeno es complejo, la generalización podría ser decepcionante.

**SEGUNDA:** Potenciar el programa agregando nuevas líneas de código que puedan asignar nuevas variables de entrada como los indicadores de mes, temperatura, punto de congelación, salinidad, refrigerante, capacidad de la planta, entre otras. Para obtener una mayor precisión en la predicción de acuerdo a los cambios en el entorno.

**TERCERA:** Comparar la red neuronal multicapa, con otros modelos, como las redes de alto orden, las redes neuronales probabilísticas, las redes neuronales difusas, las redes neuronales basadas en Wavelets o redes funcionales. Sometiendo a pruebas de predicción de series de tiempo, así como su facilidad de construcción y diseño.

## REFERENCIAS BIBLIOGRÁFICAS

- Arbib, M. A. (2003). *The Handbook of Brain Theory and Neural Networks*. Massachusetts: MIT Press Cambridge.
- Armstrong, J. S., & Green, K. C. (2006). *Demand Forecasting: Evidencebased Methods*. Monash University: Department of Econometrics and Business Statistics.
- Bigus, J. P. (1996). *Data Mining with Neural Networks*. New York: McGraw-Hill.
- Departamento de Pesca y Acuicultura. (s.f.). *La fabricación de hielo*. Roma: Organización de las Naciones Unidas para la Alimentación y la Agricultura.
- Dunne, R. A. (2007). *A Statistical Approach to Neural Networks for Pattern Recognition*. U.S.: John Wiley & Sons, Inc.
- Fausett, L. V. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall.
- Freeman, A. J., & Skapura, D. M. (1991). *Neural Networks Algorithms, Applications, and Programming Techniques*. U.S.: Addison-Wesley.
- Fyfe, C. (2000). *Artificial Neural Networks and Information Theory*. Reino Unido: The University of Paisley.
- Galushkin, A. I. (2007). *Neural Networks Theory*. Berlin: Springer-Verlag Berlin Heidelberg.
- Gómez López, W. (2010). *Pronóstico de una serie temporal usando redes neuronales*. México: Universidad Tecnológica de la Mixteca.
- Graupe, D. (2007). *Principles of Artificial Neural Networks*. Singapur: World Scientific

- Hagan, M. T., Demuth, H. B., & Beale, M. (1996). *Neural Network Design*. Beijing: PWS.
- Haykin, S. (1999). *Neural Network: A Comprehensive Foundation*. Delhi: Prentice-Hall.
- Heaton, J. (2008). *Introduction to Neural Networks for Java*. St. Louis: Heaton Research, Inc.
- Heaton, J. (2008). *Introduction to Neural Networks with C#*. U.S.: Heaton Research, Inc.
- Heaton, J. (2010). *Programming Neural Networks with Encog 2 in Java*. St. Louis: Heaton Research, Inc.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). *Metodología de la Investigación*. México: McGRAW - HILL.
- Hu, Y. H., & Hwang, J.-N. (2002). *Handbook of Neural Network Signal Processing*. U.S.: CRC Press LLC.
- Iriondo, M. H. (2007). *Introducción a la Geología*. Córdoba: Editorial Brujas.
- Isasi Viñuela, P., & Galván León, I. M. (2004). *Redes de neuronas artificiales: Un enfoque práctico*. España: Pearson.
- Jiménez Panta, J. M. (2013). *Predicción de la demanda de agua con redes neuronales artificiales en zonas urbanas de Andalucía*. Perú: Universidad de Piura.
- Khare, M., & Nagendra Shiva, S. M. (2007). *Artificial Neural Networks in Vehicular Pollution Modelling*. Berlin: Springer-Verlag Berlin Heidelberg.
- Krose, B., & van der Smagt, P. (1996). *An introduction to neural networks*. Amsterdam: University of Amsterdam.

- Mehrotra, K., Mohan, C. K., & Ranka, S. (1997). *Elements of Artificial Neural Networks*. Massachusetts: MIT Press Cambridge.
- Mendoza, L. D. (2011). *Uso de la tecnología como herramienta competitiva en la comercialización de la producción artesanal*. Ecuador: Universidad estatal de Milagro.
- Nojek, S., Britos, P., Brossi, B., & Garcia-Martinez, R. (2003). Pronóstico de Ventas: Comparación de Predicción basada en Redes Neuronales versus Método Estadístico. *Reportes Técnicos en Ingeniería del Software*, 12.
- Parks, R. W., Levine, D. S., & Long, D. L. (1998). *Fundamentals of neural network modeling: neuropsychology and cognitive neuroscience*. Massachusetts: MIT Press Cambridge.
- Quinlan, P. T. (2003). *Connectionist Models of Development*. New York: Psychology Press.
- Rabuñal, J. R., & Dorado, J. (2006). *Artificial Neural Networks in Real-Life Applications*. Reino Unido: Idea Group Inc.
- Rao, V. B., & Rao, H. (1995). *C++ Neural Networks and Fuzzy Logic*. U.S.: M & T Books.
- Rojas, R. (1996). *Neural Networks A Systematic Introduction*. Berlin: Springer-Verlag Berlin Heidelberg.
- Sapuan, S. M., & Mujtaba, I. M. (2010). *Composite Materials Technology Neural Network Applications*. U.S.: Taylor and Francis Group.
- Sarangapani, J. (2006). *Neural network control of nonlinear discrete-time systems*. Rolla, Missouri: Taylor & Francis Group.
- Taylor, B. J. (2006). *Methods and procedures for the verification and validation of Artificial Neural Networks*. U.S.: Springer Science Business Media, Inc.

- Tirozzi, B., Puca, S., Pittalis, S., Bruschi, A., Morucci, S., Ferraro, E., & Corsini, S. (2006). *Neural Networks and Sea Time Series*. Boston: Birkhauser Boston.
- Universidad de Texas. (2001). *XV Censo industrial: productos y materias primas. Industrias manufactureras*. México: Instituto Nacional de Estadística, Geografía e Informática.
- Veelenturf, L. (1995). *Analysis and Applications of Artificial Neural Networks*. Reino Unido: Prentice-Hall.
- Yanarico Espinoza, W. S. (2013). *Desarrollo de un software utilizando redes neuronales artificiales para la simulación del ciclo hidrológico de la laguna Aricota*. Perú: Universidad Nacional Jorge Basadre Grohmann.
- Yeung, D. S., Cloete, I., Shi, D., & Ng, W. W. (2010). *Sensitivity Analysis for Neural Networks*. New York: Springer-Verlag Berlin Heidelberg.
- Zhang, G. P. (2004). *Neural Networks in Business Forecasting*. Georgia: Idea Group Inc.
- Zurada, Z. M. (1992). *Introduction to Artificial Neural Systems*. Saint Paul, Minnesota: West.

## ANEXOS

### Anexo 01. Matriz de Consistencia

**Título**

Uso de redes neuronales artificiales para el pronóstico de demanda de hielo industrial en Hielos Santa Lucía S.R.L. - 2015

Problema	Objetivos	Hipótesis	Variables
<p style="text-align: center;"><b>General</b></p> <p>¿Es factible utilizar redes neuronales artificiales para predecir la demanda de hielo industrial en Hielos Santa Lucía S.R.L.?</p>	<p style="text-align: center;"><b>GENERAL</b></p> <p>Predecir la demanda de hielo industrial en Hielos Santa Lucía S.R.L., utilizando redes neuronales artificiales</p>	<p style="text-align: center;"><b>GENERAL</b></p> <p>H<sub>0</sub>: Las redes neuronales artificiales no predicen la demanda de hielo industrial en Hielos Santa Lucía S.R.L. H<sub>1</sub>: Las redes neuronales artificiales predicen la demanda de hielo industrial en Hielos Santa Lucía S.R.L.</p>	<p style="text-align: center;"><b>Variable independiente</b></p> <p>Redes neuronales artificiales.</p> <p style="text-align: center;"><b>Indicadores</b></p> <ul style="list-style-type: none"> <li>• Valor del test validación.</li> <li>• Valor del error cuadrático medio.</li> </ul>
<p style="text-align: center;"><b>Específicas</b></p> <p><b>Problema 1</b> ¿Es realizable diseñar el modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.?</p> <p><b>Problema 2</b> ¿Es posible calcular valores futuros a partir del modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.?</p>	<p style="text-align: center;"><b>Específicas</b></p> <p><b>Objetivo 1</b> Diseñar el modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.</p> <p><b>Objetivo 2</b> Calcular valores futuros a partir del modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.</p>	<p style="text-align: center;"><b>Específicas</b></p> <p><b>Subhipótesis 1</b> H<sub>0</sub>: Si en la construcción no seguimos los pasos que componen el proceso de aprendizaje. Entonces, no es realizable diseñar el modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L. H<sub>1</sub>: Si en la construcción seguimos los pasos que componen el proceso de aprendizaje. Entonces, es realizable diseñar el modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.</p> <p><b>Subhipótesis 2</b> H<sub>0</sub>: Si en el proceso de aprendizaje no se estabiliza el error de entrenamiento. Entonces, no es posible calcular valores futuros a partir del modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L. H<sub>1</sub>: Si en el proceso de aprendizaje se estabiliza el error de entrenamiento. Entonces, es posible calcular valores futuros a partir del modelo de predicción de demanda de hielo industrial en Hielos Santa Lucía S.R.L.</p>	<p style="text-align: center;"><b>Variable dependiente</b></p> <p>Demanda de hielo industrial.</p> <p style="text-align: center;"><b>Indicadores</b></p> <ul style="list-style-type: none"> <li>• Cantidad.</li> <li>• Valor máximo.</li> <li>• Valor mínimo.</li> </ul>

**PLANTA DE HIELO DE 40 TM/DÍA  
COMPRESORES DE TORNILLO BITZER  
ESPECIFICACIONES TÉCNICAS**

**A: COMPONENTES DEL SISTEMA FRIGORÍFICO:**

Los componentes del sistema frigorífico son:

1. Dos (02) unidades de compresión, c/u compuesto de:

- 01 Compresor de amoníaco BITZER (Germany),  
Modelo : OSKA 7471-K  
Tipo de compresor : Tornillo  
Capacidad : 43,5 TR (153,0 kW)  
Velocidad : 3,500 RPM  
Condiciones de operación : -15°C succión y +35°C de condensación  
Transmisión por acoplamiento directo  
Control para el arranque descargado y 50% - 75% de control de capacidad.

• 01 Motor eléctrico de 75 HP a 3500 RPM:

440/220 V, 60 Hz y 3 fases; factor de servicio 1,15

Marca Siemens (ensamblado en Colombia)

Se incluye un sistema electrónico de protección INT 69 VS (ver esquema eléctrico)

• 01 conjunto de controles y accesorios para inyección de aceite al compresor como son:

Solenoides, filtro, control de flujo de aceite, visor de aceite, etc. (ver esquema Bitzer adjunto)

Cada compresor viene de fábrica para ser acoplado a un motor eléctrico, mediante un acople KS 730 y una brida de fijación GS 7120, tal como se ve en la figura de la página 9 del catálogo Bitzer.

Para cada unidad de compresión, la fábrica envía adicionalmente lo siguiente:

- Un separador de aceite principal, tipo OA 4088 a, ver página 25 del manual Bitzer.  
Viene con control de nivel de aceite, control electrónico de nivel mínimo de aceite, visor de vidrio para nivel de aceite, calentador de aceite, termostato, conexión para válvula de seguridad. (Ver plano OA 4088)
- Un separador de aceite secundario, tipo OAS 1655 (ver posición 13, del esquema Bitzer), ver página 11 del manual Bitzer.
- Un enfriador de aceite, tipo OW 290 A, enfriado por agua, las especificaciones de este enfriador se encuentra en la hoja técnica Bitzer.

Todos estos componentes se instalan sobre una base metálica que debe ser preparada por nosotros, bajo las recomendaciones de la fábrica Bitzer.

2. Un (01) condensador evaporativo, de las siguientes características:

Fabricante	: Mebrafe (Brasil)
Modelo	: CETF – 390
Capacidad	: 130 TR (396,000 kcal/h)
Temperatura de condensación	: +36°C
Temperatura de bulbo húmedo	: +24°C
Cantidad/CV ventiladores axiales	: 3 x 2 CV
Flujo de aire en m <sup>3</sup> /h	: 42,000
Cantidad/CV bomba de agua	: 1 x 3 CV
Flujo de agua, en m <sup>3</sup> /h	: 52,0
Dimensiones	: 3270 x 1800 x 2670(h)
Peso en vacío, en Kg	: 2768
Peso en operación, Kg	: 4239

3. Un (01) tanque recibidor de líquido de NH<sub>3</sub>, de alta presión:
  - Fabricante : MITOR
  - Tipo : Horizontal
  - Capacidad : 1,200 litros
  - Dimensiones : 24"Ø x 12' longitud
  - Espesor de pared : 5/16"
4. Dos (02) tanques acumuladores de succión:
  - Fabricante : MITOR
  - Tipo : Vertical
  - Capacidad : 220 litros
  - Dimensiones : 18" Ø x 4' alto
5. Dos (02) enfriadores de salmuera:
  - Fabricante : MITOR
  - Tipo : Industrial, forma de serpentín
  - Capacidad frigorífica de c/u : 40 TR (120,960 Kcal/hr)
  - Material de fabricación : Tubo de acero galvanizado, de 1' Ø ISO-1
6. Un (01) desmoldeador de bloques, del tipo giratorio.
7. Dos (02) sistema de grúa puente móvil, para traslado de los bloques.
8. Dos (02) moledoras de hielo de 15 TM/hr de capacidad, con motor de 30 HP.

#### RESERVAS:

En la presente oferta no se incluye lo siguiente:

- a) Tanque de concreto para instalar el tanque de salmuera.
- b) Tablones de madera para las tapas de los tanques de salmuera.
- c) Bases de cimentación, obras civiles, etc.
- d) Refrigerante R-717 (NH<sub>3</sub>), ni sal industrial para la salmuera.
- e) Transporte de los materiales de fabricación.
- f) Ningún otro componente no detallado en la presente oferta.
- g) Instalación de lo ofertado, porque no conocemos la disposición de la planta.

Nota: Los gastos de transporte y estadía de nuestro personal es por cuenta del cliente.

### INSTRUCCIONES DE OPERACIÓN

#### **Control de seguridad:**

La unidad de compresión Bitzer tiene los siguientes controles de seguridad:

- En la línea de aceite:
  - o Control de nivel en el separador de aceite.
  - o Termostato de aceite en el separador.
  - o Control de flujo de aceite.
- En la línea de succión: Presostato de baja, Danfoss KP1A
- En la línea de descarga: Presostato de alta, Danfoss KP5A
- Válvula de seguridad en el separador de aceite.

El tanque recipiente de líquido tiene una válvula dual de seguridad, que funciona con una válvula de tres vías; esta válvula solamente controla a una válvula de seguridad, en caso de que actúe se cierra y se abre para la otra válvula.

El evaporador tiene un tanque separador de líquido que tiene dos controles de nivel:

El primero actúa directamente sobre la válvula solenoide que permite un constante flujo de amoníaco al sistema; el segundo actúa directamente sobre el compresor, apagando el sistema porque el nivel de NH<sub>3</sub> ha subido por encima del permitido. Además el tanque lleva una válvula de seguridad de baja presión.

### **Control de capacidad:**

Cada compresor tiene dos presostatos Penn que permite controlar el funcionamiento de las válvulas solenoides; al bajar la presión de succión por debajo de la establecida, el primer presostato desenergiza a la bobina del solenoide y el compresor funciona a 75%; si sigue bajando y el compresor funciona al 50%.

La presión regulada en el primer presostato es de 17 psig (abre contacto) y 25 psig (cierre contacto)  
La presión regulada en el segundo presostato es de 16 psig (abre contacto) y 24 psig (cierra contacto)

### **Funcionamiento del circuito frigorífico:**

El circuito ha sido diseñado para funcionar en las siguientes condiciones:

Temperatura de evaporación	: -15°C
Temperatura de condensación	: +35°C
Presión de evaporación	: 18 psig
Presión de condensación	: 180 psig

El compresor succiona los gases de amoníaco que se encuentra en el evaporador a una presión de 20# y lo comprime hasta 80#; el gas de alta presión ingresa al condensador evaporativo donde se condensa y en forma de líquido se acumula en el tanque recipiente de líquido. La cantidad de refrigerante se puede observar por medio de un visor de nivel de líquido.

Si la válvula solenoide del evaporador se encuentra energizada, significa que necesita refrigerante, entonces por diferencia de presiones, el amoníaco líquido llega desde el recipiente de alta hasta el tanque separador de líquido. Antes de ingresar al tanque, el líquido que se encuentra a 180# pasa por medio de una válvula de expansión manual bajando la presión y temperatura e ingresa una mezcla de líquido y vapor a -15°C; el líquido sigue su camino hacia los tubos del evaporador que finalmente es succionado por el compresor.

### **Modo de operar con el compresor:** puesta en marcha del compresor

- Prender el compresor evaporativo.
- Prender el compresor en 50% manual; poner en ON el solenoide de líquido.
- Abrir la válvula 2<sup>1</sup>/<sub>2</sub> que se encuentra en el evaporador poco a poco, sin sobrepasar la presión de 50#.
- Una vez abierta la válvula y con presión de succión menor a 50#, aumentar la capacidad del compresor al 100%.
- Verificar nivel de aceite y temperatura del mismo.

Parada del compresor:

- Poner en OFF la válvula solenoide de líquido.
- Esperar que recoja un poco de refrigerante y descargue al 50% y entonces apagar el compresor.
- Cerrar la válvula de 2<sup>1</sup>/<sub>2</sub> del evaporador.
- Apagar el condensador.

### **Cómo operar sistema de refrigeración usando compresores Bitzer:**

Los mecánicos de mantenimiento deberán de tomar en cuenta que el servicio a las plantas de refrigeración y compresores deberá ser continuo, o sea, lo que podemos llamar un servicio preventivo, si esto no se hace, se tendrá que dar servicios correctivos, que son muy costosos en la mayor parte de las ocasiones, y provocan paros injustificados que muchas veces ponen en peligro la estabilidad de los productos que se manejan.

Recomendamos observar que el calentamiento de los motores eléctricos siempre esté dentro de los límites establecidos por los fabricantes. Revisar cada 2 o 3 meses que los controles automáticos para

alta presión y baja presión se encuentren funcionando adecuadamente, chequear los acoplamientos no se aflojen y al mismo tiempo deberá chequearse que el retorno de aceite sea el correcto. Recomendaciones para el mantenimiento preventivo de los compresores de tornillo se entrega junto a este manual.

Por lo menos una vez al mes se deberá purgar aceite en todas las válvulas instaladas para este propósito.

#### **Cómo hacer la purga de los gases no condensables y el aire en el sistema:**

Es muy importante que los sistemas de amoníaco operen sin gases no condensables o aire dentro del sistema con el objeto de bajos costos de energía.

Entre más alta sea la presión de descarga más elevado será el costo de la energía eléctrica.

Se sabe que a la presión del amoníaco líquido saturado o sea cuando sale del condensador corresponde una determinada temperatura de acuerdo con la tabla siguiente:

Presión del NH <sub>3</sub> a la salida del condensador	Temperatura correspondiente
150 psig	29,11°C
180 psig	34,83°C
210 psig	39,88°C

Cuando se encuentre que una planta tiene refrigerante líquido saliendo del condensador a una temperatura más alta que la que obtenemos en la tabla anterior, quiere decir que hay gases extraños en el sistema y que debemos eliminarlos.

Procedimiento manual de purga de gases:

1. Pare el compresor
2. Deje que el agua siga pasando por el condensador
3. Cierre la válvula de succión y descarga del compresor
4. Espere ½ hora aproximadamente hasta que el líquido haya pasado del condensador al recipiente de líquido.
5. Coloque una manguera, primero en la válvula de purga que se encuentra arriba del condensador, con el otro extremo de un recipiente lleno de agua.
6. Abra un poco la válvula de purga y los gases no condensables pasarán por la manguera al agua.
7. Cuando se oiga un ruido en el recipiente de agua, quiere decir que está saliendo amoníaco por la manguera, cierre la válvula y espere de 15 a 20 minutos, volviendo a abrir después de este tiempo un poco de válvula; si se vuelve a oír el ruido en el agua donde se encuentra la manguera se puede considerar que el sistema está libre de gases no condensables, cierre la válvula de purga.
8. Arranque el compresor.

#### **Incrustaciones en el condensador evaporativo:**

Es necesario purgar el agua del condensador a fin de controlar la concentración de los sólidos disueltos. Si la purga no se efectúa en forma adecuada, ocurre formación de depósitos en los tubos. Los tubos del condensador que son de acero galvanizado no se debe emplear el ácido clorhídrico para la limpieza.

Con agua de buena calidad, la purga es aproximadamente 1.1 l/h por kw de calor disipado; para el caso son 502 litros por hora (1/2 m<sup>3</sup> por hora), que se debe regular con la boya de alimentación de agua.

Se ha recopilado información de Isasi Viñuela & Galván León (2004).

**Propagación de los patrones de entrada**

- Activación de las neuronas de la capa de entrada ( $a_i^1$ ). Las neuronas de la capa de entrada se encargan de transmitir, hacia la red, las señales recibidas del exterior. Por tanto:

**Fórmula 1.1**

$$a_n^1 = x_i \text{ para } i = 1, 2, \dots, n_1$$

Donde  $X = (x_1, x_2, \dots, x_{n_1})$  representa el vector o patrón de entrada a la red.

- Activación de las neuronas de la capa oculta  $C(a_i^C)$ . Las neuronas ocultas de la red procesan la información recibida aplicando la función de activación  $f$  a la suma de los productos de las activaciones que recibe por sus correspondientes pesos, es decir:

**Fórmula 1.2**

$$a_i^C = f \left( \sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) \text{ para } i = 1, 2, \dots, n_C \text{ y } C = 2, 3, \dots, C - 1$$

Donde  $a_i^{C-1}$  son las activaciones de las neuronas de la capa  $C - 1$ .

- Activación de las neuronas de capa de salida ( $a_i^C$ ). Al igual que en el caso anterior, la activación de estas neuronas viene dada por la función de activación  $f$  aplicada a la suma de los productos de las entradas que recibe por sus correspondientes pesos:

**Fórmula 1.3**

$$y_i = a_i^C = f \left( \sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) \text{ para } i = 1, 2, \dots, n_C$$

Donde  $Y = (y_1, y_2, \dots, y_{n_C})$  es el vector salida de la red.

La función  $f$  es la llamada función de activación. Para el perceptrón multicapa, las funciones de activación más utilizadas son la función sigmoideal y la función tangente hiperbólica. Dichas funciones poseen como imagen un rango continuo de valores dentro de los intervalos  $[0,1]$  y  $[-1,1]$ , respectivamente, y vienen dadas por las siguientes expresiones:

- Función sigmoideal:

**Fórmula 1.5**

$$f_1(x) = \frac{1}{1 + e^{-x}}$$

- Función tangente hiperbólica:  
**Fórmula 1.6**

$$f_2(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

### Algoritmo de Back Propagation

#### Fórmula 1.7

$$\text{Min}_w E$$

#### Fórmula 1.8

$$E = \frac{1}{N} \sum_{n=1}^N e(n)$$

#### Fórmula 1.9

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2$$

### Regla delta generalizada

- Pesos de la capa oculta C-1 a la capa de salida y umbrales de la capa de salida

Pesos:

#### Fórmula 1.10

$$w_{ji}^{c-1}(n) = w_{ji}^{c-1}(n-1) + \alpha \delta_i^c(n) a_j^{c-1}(n)$$

$$\text{para } j = 1, 2, \dots, n_{c-1} \text{ y } i = 1, 2, \dots, n_c$$

Umbrales:

#### Fórmula 1.11

$$u_i^c(n) = u_i^c(n-1) + \alpha \delta_i^c(n)$$

$$\text{para } i = 1, 2, \dots, n_c$$

Donde:

#### Fórmula 1.12

$$\delta_i^c(n) = -(s_i(n) - y_i(n)) y_i(n) (1 - y_i(n))$$

- Pesos de la capa c a la capa c+1 y umbrales de las neuronas de la capa c+1 para c=1,2,...c-2

Pesos:

**Fórmula 1.13**

$$w_{kj}^c(n) = w_{kj}^c(n-1) + \alpha \delta_i^{c+1}(n) a_k^c(n)$$

para  $j = 1, 2, \dots, n_{c+1}$  y  $C = 1, 2, \dots, C-2$

Umbrales:

**Fórmula 1.14**

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) + \alpha \delta_i^{c+1}(n)$$

para  $j = 1, 2, \dots, n_{c+1}$  y  $C = 1, 2, \dots, C-2$

Donde:

**Fórmula 1.15**

$$\delta_j^{c+1}(n) = a_j^c(n)(1 - a_j^c(n)) \sum_{i=1}^{n_{c+1}} \delta_i^{c+2}(n) w_{ji}^c$$

**Razón de aprendizaje. Inclusión del momento en la ley de aprendizaje**

**Fórmula 1.16**

$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w} + \eta \Delta w(n-1)$$

$$\Delta w(n-1) = w(n-1) - w(n-2) = -\alpha \frac{\partial e(n-1)}{\partial w} + \eta \Delta w(n-2)$$

$$= \dots = -\alpha \sum_{t=0}^{n-1} \eta^{n-1-t} \frac{\partial e(t)}{\partial w}$$

**Fórmula 1.17**

$$w(n) = w(n-1) - \alpha \sum_{t=0}^{n-1} \eta^{n-t} \frac{\partial e(t)}{\partial w}$$

## Fórmulas complementarias

### Fórmula 1.18

$$\delta_j^c(n) = -(s_i(n) - y_i(n))f' \left( \sum_{j=1}^{n_{C-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right)$$

### Fórmula 1.19

$$w_{ji}^{c-1}(n) = w_{ji}^{c-1}(n-1) + \alpha \delta_i^c(n) a_j^{c-1}(n)$$

$$\text{para } j = 1, 2, \dots, n_{C-1} \quad i = 1, 2, \dots, n_C$$

### Fórmula 1.20

$$u_i^c(n) = u_i^c(n-1) + \alpha \delta_i^c(n) \text{ para } i = 1, 2, \dots, n_C$$

### Fórmula 1.21

$$w_{kj}^c(n) = w_{kj}^c(n-1) + \alpha \delta_i^{c+1}(n) a_k^c(n)$$

$$\text{para } k = 1, 2, \dots, n_C, j = 1, 2, \dots, n_{C+1} \text{ y } C = 1, 2, \dots, C-2$$

### Fórmula 1.22

$$\delta_i^{c+1}(n) = f' \left( \sum_{k=1}^{n_C} w_{kj}^c a_k^c + u_j^c \right) \sum_{i=1}^{n_{C+1}} \delta_i^{c+2}(n) w_{ji}^c$$

### Fórmula 1.23

$$z = \frac{x - \tilde{x}}{s}$$

## Proceso de aprendizaje del perceptrón multicapa

Los pasos que componen el proceso de aprendizaje del perceptrón multicapa son los siguientes:

**Paso 1:** Se inicializan los pesos y umbrales de la red. Generalmente, esta inicialización es aleatoria y con valores alrededor del cero.

**Paso 2:** Se toma un patrón  $n$  del conjunto de entrenamiento  $(X(n), S(n))$ , y se propaga hacia la salida de la red el vector de entrada  $X(n)$  utilizando *Fórmula 1.1*, *Fórmula 1.2*, y *Fórmula 1.3* obteniéndose así la respuesta de la red para dicho vector de entrada,  $Y(n)$ .

**Paso 3:** Se evalúa el error cuadrático cometido por la red para el patrón  $n$  utilizando *Fórmula 1.9*.

**Paso 4:** Se aplica la regla delta generalizada para modificar los pesos y umbrales de la red. Para ello se siguen los siguientes pasos:

4.1 Se calculan los valores  $\delta$  para todas las neuronas de la capa de salida utilizando *Fórmula 1.18*.

4.2 Se calculan los valores  $\delta$  para el resto de neuronas de la red utilizando *Fórmula 1.22* empezando desde la última capa oculta y retro propagando dichos valores hacia la capa de entrada.

4.3 Se modifican pesos y umbrales de la red siguiendo *Fórmula 1.19* y *Fórmula 1.20* para los pesos y umbrales de la capa de salida y *Fórmula 1.21* y *Fórmula 1.22* para el resto de los parámetros de la red.

**Paso 5:** Se repiten los pasos 2,3 y 4 para todos los patrones de entrenamiento, completando así una iteración o etapa de aprendizaje.

**Paso 6:** Se evalúa el error total  $E$  de la etapa *Fórmula 1.8* cometido por la red. Dicho error también recibe el nombre de error de entrenamiento, pues se calcula utilizando los patrones de entrenamiento.

**Paso 7:** Se repiten los pasos 2, 3, 4, 5 y 6 hasta alcanzar un mínimo del error de entrenamiento, para lo cual se realizan  $m$  etapas de aprendizaje. Se sabe que llegamos a un mínimo graficando el error  $E$  de cada etapa y comprobando que se acerca asintóticamente a cero.

Anexo 04. Código MATLAB

```
function [w,u,cambio_peso,cambio_umbral] =
actualizacion(capas,a,s,w,u,alfa,eta,cambio_peso,cambio_umbral)
delta{capas} = a{capas}.*(1-a{capas}).*(a{capas} - s);
for c = capas-1 : -1 : 2
delta{c} = a{c}.*(1-a{c}).*(w{c} * (delta{c+1}))';
end
for c=1:capas-1
w{c} = w{c} - alfa * kron(delta{c+1},a{c}') + eta * cambio_peso{c} ;
cambio_peso{c} = - alfa * kron(delta{c+1},a{c}') + eta *
cambio_peso{c};
end
for c=2:capas
u{c} = u{c} - alfa * delta{c} + eta * cambio_umbral{c} ;
cambio_umbral{c} = - alfa * delta{c} + eta * cambio_umbral{c};
end
end
```

```
function [w,u] = aleatorio(n)
capas = size(n,2);
w = {}; u = {};
for c=1:capas
if c < capas
w{c} = 0.1*randn(n(c),n(c+1));
end
if c>1
u{c} = 0.1*randn(1,n(c));
end
end
end
```

```
function [input,output,INPUT_max,INPUT_min,OUTPUT_max,OUTPUT_min] =
escalamiento(INPUT,OUTPUT)
INPUT_max = max(INPUT);
INPUT_min = min(INPUT);
OUTPUT_max = max(OUTPUT);
OUTPUT_min = min(OUTPUT);
input=[];
for i=1:size(INPUT,2)
input = [input, (INPUT(:,i)-INPUT_min(i))/(INPUT_max(i)-INPUT_min(i))];
end
output=[];
for i=1:size(OUTPUT,2)
output = [output, (OUTPUT(:,i)-OUTPUT_min(i))/(OUTPUT_max(i)-
OUTPUT_min(i))];
end
end
```

```
function [INPUT,OUTPUT,capas,n,N] = excel(nombre)
datos_excel = xlsread(nombre);
capas = datos_excel(1,1);
n = datos_excel(2,1:capas);
N = datos_excel(3,1);
```

```

INPUT=datos_excel(4:4+N-1,1:n(1));
OUTPUT=datos_excel(4:4+N-1,n(1)+1:n(1)+n(capas));
end

```

```

clc; clear;
nombre = 'Normalizado';
nombre_load = strcat(nombre, '.xlsx');
serie = xlsread(nombre_load);
N = size(serie,1);
longitud_ventana = 16;
patron_fila = [];
patron = [];
paso = 0;
while paso < N-longitud_ventana
for t=1:longitud_ventana
patron_fila = [ patron_fila , serie(paso + t)];
end
patron_fila = [patron_fila , serie(paso + t + 1)];
paso = paso + 1;
patron = [patron;patron_fila];
patron_fila = [];
end
nuevo_nombre = strcat(nombre, '_patrones.xlsx');
n = [longitud_ventana,8,1];
xlswrite(nuevo_nombre,size(n,2), 'Hojal', 'A1')
xlswrite(nuevo_nombre,n, 'Hojal', 'A2')
xlswrite(nuevo_nombre,size(patron,1), 'Hojal', 'A3')
xlswrite(nuevo_nombre,patron, 'Hojal', 'A4')

```

```

clc; clear
nombre = 'Normalizado_patrones.xlsx';
[INPUT,OUTPUT,capas,n,N] = excel(nombre);
[input,output,INPUT_max,INPUT_min,OUTPUT_max,OUTPUT_min] =
escalamiento(INPUT,OUTPUT);
[w,u] = aleatorio(n);
alfa = 0.02;
eta = 0.99;
[cambio_peso,cambio_umbral] = reset_momento(n);
for etapa=1:250
for p=1:N/2
x = input(p, :);
s = output(p, :);
[a,y] = salida_red(x,w,u);
e(etapa,p) = 0.5 * dot(s-y,s-y);
[w,u,cambio_peso,cambio_umbral] =
actualizacion(capas,a,s,w,u,alfa,eta,cambio_peso,cambio_umbral);
end
e_media(etapa) = (1/(N/2)) * sum(e(etapa,:));
end
plot(e_media)
validacion = valida(N,input,output,w,u);
disp(['Test de validación = ',num2str(validacion),'%']);
save optimol w u INPUT_max INPUT_min OUTPUT_max OUTPUT_min

```

```

function [cambio_peso,cambio_umbral] = reset_momento(n)
capas = size(n,2);

```

```
for c=1:capas
if c < capas
cambio_peso{c} = zeros(n(c),n(c+1));
end
if c>1
cambio_umbral{c} = zeros(1,n(c));
end
end
```

```
function [cambio_peso,cambio_umbral] = reset_momento(n)
capas = size(n,2);
for c=1:capas
if c < capas
cambio_peso{c} = zeros(n(c),n(c+1));
end
if c>1
cambio_umbral{c} = zeros(1,n(c));
end
end
```

```
function tanto_por_ciento = valida(N,input,output,w,u)
exito = 0;
for p = (N/2)+1 : N
x = input(p,:);
s = output(p,:);
[a,y] = salida_red(x,w,u);
if 0.5 * dot(s-y,s-y) < 0.005
exito = exito + 1;
end
end
tanto_por_ciento = 100*(exito/(N/2));
end
```

```
clc;clear
nombre = 'Normalizado';
nombre_load = strcat(nombre, '.xlsx');
serie = xlsread(nombre_load);
N = size(serie,1);
longitud_ventana = 16;
load optimol;
inicio = 1;
final = inicio + longitud_ventana - 1;
x = (serie(inicio:final))';
[a,y] = salida_red(x,w,u);
s = serie(final + 1);
error_relativo = 100*abs(s-y)/s
```

## Anexo 05. Resultados

>>INPUT\_max <1x16 double>

	1	2	3	4
1	0,885083457010999	0,885083457010999	0,885083457010999	0,885083457010999

	5	6	7	8
1	0,885083457010999	0,885083457010999	0,885083457010999	0,885083457010999

	9	10	11	12	13	14	15	16
1	0,885083457010999	0,989631406672495	1	1	1	1	1	1

>>INPUT\_min <1x16 double>

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0

	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0

>>N <1x1 double>

	1
1	50

>>OUTPUT <50x1 double>

1	0,402504726858723
2	0,385549027181979
3	0,184621952955049
4	0,0194360298452843
5	0,0156138817167138
6	0,122329070689410
7	0,254953544635778
8	0,690649968487609
9	0,843161811047635
10	0,365226584260069
11	0,349035314209039
12	0,210584097424116
13	0,254579462052982
14	0,253034338341432
15	0,0870555228007400
16	0,00461504055949743
17	0
18	0,104438164555675
19	0,302661272287393
20	0,538048671396914
21	0,675751723016244
22	0,432081647589810
23	0,154467643889646
24	0,0940777034582308
25	0,194441620753451
26	0,292150364933824
27	0,134242787728465
28	0,0616423038607762
29	0,0674771789293919
30	0,210978510582064
31	0,766869294732348
32	0,885083457010999
33	0,828962937361498
34	0,458440238274341
35	0,178378433325879
36	0,315064956187611
37	0,407699188810051
38	0,236881696383191
39	0,0535812308130197
40	0,0340435480919755
41	0,0552686685906333
42	0,457193974017525
43	0,989631406672495
44	1
45	0,902494561571147
46	0,280338300770529
47	0,226004838676886
48	0,211702279057475
49	0,181735011283469
50	0,158171874682335

>>OUTPUT\_max <1x1 double>

	1
1	1

>>OUTPUT\_min <1x1 double>

	1
1	0

>>a <1x3 cell>

	1
1	<1x16 double>

2	
[0,407584240013181,0,801342270437964,0,585648135599687,0,368873947720997,0,187852532630902,0,750498020501026,0,204972922973021,0,242449680646578]	

3	
0,230516672000345	

a{1}

	1	2	3	4	5	6	7	8
1	0,9526	0,4126	0,3944	0,2379	0,2876	0,2859	0,0984	0,0052

	9	10	11	12	13	14	15	16
1	0	0,1055	0,3027	0,5380	0,6758	0,4321	0,1545	0,0941

a{2}

	9	10	11	12	13	14	15	16
1	0,4076	0,8013	0,5856	0,3689	0,1879	0,7505	0,2050	0,2424

a{3}

	1
1	0,2305

>>alfa <1x1 double>

	1
1	0,0200

>>ans <8x1 double>

	1
1	-1,23350445339543e-05
2	-7,43404001276172e-05
3	1,41242895144283e-05
4	1,22199767607642e-06
5	-0,000182350903191513
6	2,20830156768497e-05
7	-0,000149272962285427
8	-4,70569933451120e-05

>>cambio\_peso <1x2 cell>

	1
1	<16x8 double>

2	
[-1,23350445339543e-05;-7,43404001276172e-05;1,41242895144283e-05;1,22199767607642e-06;-0,000182350903191513;2,20830156768497e-05;-0,000149272962285427;-4,70569933451120e-05]	

>>cambio\_umbral <1x3 cell>

	1
1	[]

2	
[9,24644573240136e-05,0,000147429031574665,-4,74509123935619e-05,4,68090420254537e-05,0,000111589916142412,-7,88491054338906e-06,8,7988969012209e-05,8,06097926281307e-06]	

3	
1,88166742656567e-06	

>>capas <1x1 double>

	1
1	3

>>e

	1
1	<250x25 double>

>>e media

	1
1	<1x250 double>

>>eta <1x1 double>

	1
1	0,9900

>>etapa <1x1 double>

	1
1	250

>>input

	1
1	<50x16 double>

>>n <1x3 double>

	1	2	3
1	16	8	1

>>nombres <1x25 char>

	1
1	Normalizado_patrones.xls

>>output

	1
1	<50x1 double>

>>n <1x1 double>

	1
1	25

>>s <1x1 double>

	1
1	0,1944

>>n <1x3 cell>

	1
1	□

	2	3
1	[0,709036021429658,1,38626649753396,-0,278970803250810,0,651541993420849,0,613353615133087,-0,215003903830362,0,191568545691718,-0,138556974569644]	1.9177

>>s <1x1 double>

	1
1	72

>>w <1x2 cell>

	1	2
1	<16x8 double>	[-2,80443339669528,-3,32190810093663,2,15273353697362,-2,17048055479815,-2,62253662541127,1,77406216053654,-1,95771149141476,-0,890089952280075]

>>x <1x16 double>

	1	2	3	4	5	6	7	8
1	0,952635375080 970	0,412646492674 792	0,394352997385 987	0,237925695883 274	0,287633284789 570	0,285887546916 705	0,0983585469948 041	0,00521424338342 376

	9	10	11	12	13	14	15	16
1	0	0,105532386958933	0,302661272287393	0,538048671396914	0,675751723016244	0,432081647589810	0,154467643889646	0,0940777034582308





>>INPUT <50x16 double>

1	0,804342610852461	0,471144814686807	0,291955191412365	0,122573037591233	0,112261369874154	0,111228576656434	0,0809563502551487	0,234663630634111
2	0,471144814686807	0,291955191412365	0,122573037591233	0,112261369874154	0,111228576656434	0,0809563502551487	0,234663630634111	0,265740947811414
3	0,291955191412365	0,122573037591233	0,112261369874154	0,111228576656434	0,0809563502551487	0,234663630634111	0,265740947811414	0,242580356598288
4	0,122573037591233	0,112261369874154	0,111228576656434	0,0809563502551487	0,234663630634111	0,265740947811414	0,242580356598288	0,682253440949844
5	0,112261369874154	0,111228576656434	0,0809563502551487	0,234663630634111	0,265740947811414	0,242580356598288	0,682253440949844	0,576229491532315
6	0,111228576656434	0,0809563502551487	0,234663630634111	0,265740947811414	0,242580356598288	0,682253440949844	0,576229491532315	0,736658060056519
7	0,0809563502551487	0,234663630634111	0,265740947811414	0,242580356598288	0,682253440949844	0,576229491532315	0,736658060056519	0,684375952995710
8	0,234663630634111	0,265740947811414	0,242580356598288	0,682253440949844	0,576229491532315	0,736658060056519	0,684375952995710	0,630491796612926
9	0,265740947811414	0,242580356598288	0,682253440949844	0,576229491532315	0,736658060056519	0,684375952995710	0,630491796612926	0,424368227377152
10	0,242580356598288	0,682253440949844	0,576229491532315	0,736658060056519	0,684375952995710	0,630491796612926	0,424368227377152	0,402504726858723
11	0,682253440949844	0,576229491532315	0,736658060056519	0,684375952995710	0,630491796612926	0,424368227377152	0,402504726858723	0,385549027181979
12	0,576229491532315	0,736658060056519	0,684375952995710	0,630491796612926	0,424368227377152	0,402504726858723	0,385549027181979	0,184621952955049
13	0,736658060056519	0,684375952995710	0,630491796612926	0,424368227377152	0,402504726858723	0,385549027181979	0,184621952955049	0,0194360298452843
14	0,684375952995710	0,630491796612926	0,424368227377152	0,402504726858723	0,385549027181979	0,184621952955049	0,0194360298452843	0,0156138817167138
15	0,630491796612926	0,424368227377152	0,402504726858723	0,385549027181979	0,184621952955049	0,0194360298452843	0,0156138817167138	0,122329070689410
16	0,424368227377152	0,402504726858723	0,385549027181979	0,184621952955049	0,0194360298452843	0,0156138817167138	0,122329070689410	0,254953544635778
17	0,402504726858723	0,385549027181979	0,184621952955049	0,0194360298452843	0,0156138817167138	0,122329070689410	0,254953544635778	0,690649968487609
18	0,385549027181979	0,184621952955049	0,0194360298452843	0,0156138817167138	0,122329070689410	0,254953544635778	0,690649968487609	0,843161811047635
19	0,184621952955049	0,0194360298452843	0,0156138817167138	0,122329070689410	0,254953544635778	0,690649968487609	0,843161811047635	0,365226584260069
20	0,0194360298452843	0,0156138817167138	0,122329070689410	0,254953544635778	0,690649968487609	0,843161811047635	0,365226584260069	0,349035314209039
21	0,0156138817167138	0,122329070689410	0,254953544635778	0,690649968487609	0,843161811047635	0,365226584260069	0,349035314209039	0,210584097424116
22	0,122329070689410	0,254953544635778	0,690649968487609	0,843161811047635	0,365226584260069	0,349035314209039	0,210584097424116	0,254579462052982
23	0,254953544635778	0,690649968487609	0,843161811047635	0,365226584260069	0,349035314209039	0,210584097424116	0,254579462052982	0,253034338341432
24	0,690649968487609	0,843161811047635	0,365226584260069	0,349035314209039	0,210584097424116	0,254579462052982	0,253034338341432	0,0870555228007400
25	0,843161811047635	0,365226584260069	0,349035314209039	0,210584097424116	0,254579462052982	0,253034338341432	0,0870555228007400	0,00461504055949743
26	0,365226584260069	0,349035314209039	0,210584097424116	0,254579462052982	0,253034338341432	0,0870555228007400	0,00461504055949743	0
27	0,349035314209039	0,210584097424116	0,254579462052982	0,253034338341432	0,0870555228007400	0,00461504055949743	0	0,104438164555675
28	0,210584097424116	0,254579462052982	0,253034338341432	0,0870555228007400	0,00461504055949743	0	0,104438164555675	0,302661272287393
29	0,254579462052982	0,253034338341432	0,0870555228007400	0,00461504055949743	0	0,104438164555675	0,302661272287393	0,538048671396914
30	0,253034338341432	0,0870555228007400	0,00461504055949743	0	0,104438164555675	0,302661272287393	0,538048671396914	0,675751723016244
31	0,0870555228007400	0,00461504055949743	0	0,104438164555675	0,302661272287393	0,538048671396914	0,675751723016244	0,432081647589810
32	0,00461504055949743	0	0,104438164555675	0,302661272287393	0,538048671396914	0,675751723016244	0,432081647589810	0,154467643889646
33	0	0,104438164555675	0,302661272287393	0,538048671396914	0,675751723016244	0,432081647589810	0,154467643889646	0,0940777034582308
34	0,104438164555675	0,302661272287393	0,538048671396914	0,675751723016244	0,432081647589810	0,154467643889646	0,0940777034582308	0,194441620753451
35	0,302661272287393	0,538048671396914	0,675751723016244	0,432081647589810	0,154467643889646	0,0940777034582308	0,194441620753451	0,292150364933824
36	0,538048671396914	0,675751723016244	0,432081647589810	0,154467643889646	0,0940777034582308	0,194441620753451	0,292150364933824	0,134242787728465
37	0,675751723016244	0,432081647589810	0,154467643889646	0,0940777034582308	0,194441620753451	0,292150364933824	0,134242787728465	0,0616423038607762
38	0,432081647589810	0,154467643889646	0,0940777034582308	0,194441620753451	0,292150364933824	0,134242787728465	0,0616423038607762	0,0674771789293919
39	0,154467643889646	0,0940777034582308	0,194441620753451	0,292150364933824	0,134242787728465	0,0616423038607762	0,0674771789293919	0,210978510582064
40	0,0940777034582308	0,194441620753451	0,292150364933824	0,134242787728465	0,0616423038607762	0,0674771789293919	0,210978510582064	0,766869294732348
41	0,194441620753451	0,292150364933824	0,134242787728465	0,0616423038607762	0,0674771789293919	0,210978510582064	0,766869294732348	0,885083457010999
42	0,292150364933824	0,134242787728465	0,0616423038607762	0,0674771789293919	0,210978510582064	0,766869294732348	0,885083457010999	0,828962937361498
43	0,134242787728465	0,0616423038607762	0,0674771789293919	0,210978510582064	0,766869294732348	0,885083457010999	0,828962937361498	0,458440238274341
44	0,0616423038607762	0,0674771789293919	0,210978510582064	0,766869294732348	0,885083457010999	0,828962937361498	0,458440238274341	0,178378433325879
45	0,0674771789293919	0,210978510582064	0,766869294732348	0,885083457010999	0,828962937361498	0,458440238274341	0,178378433325879	0,315064956187611
46	0,210978510582064	0,766869294732348	0,885083457010999	0,828962937361498	0,458440238274341	0,178378433325879	0,315064956187611	0,407699188810051
47	0,766869294732348	0,885083457010999	0,828962937361498	0,458440238274341	0,178378433325879	0,315064956187611	0,407699188810051	0,236881696383191
48	0,885083457010999	0,828962937361498	0,458440238274341	0,178378433325879	0,315064956187611	0,407699188810051	0,236881696383191	0,0535812308130197
49	0,828962937361498	0,458440238274341	0,178378433325879	0,315064956187611	0,407699188810051	0,236881696383191	0,0535812308130197	0,0340435480919755
50	0,458440238274341	0,178378433325879	0,315064956187611	0,407699188810051	0,236881696383191	0,0535812308130197	0,0340435480919755	0,0552686685506333

	9	10	11	12	13	14	15	16
1	0.265740947811414	0.242580356598288	0.682253440949844	0.576229491532315	0.736658060056519	0.684375952995710	0.630491796612926	0.424368227377152
2	0.242580356598288	0.682253440949844	0.576229491532315	0.736658060056519	0.684375952995710	0.630491796612926	0.424368227377152	0.402504726858723
3	0.682253440949844	0.576229491532315	0.736658060056519	0.684375952995710	0.630491796612926	0.424368227377152	0.402504726858723	0.385549027181979
4	0.576229491532315	0.736658060056519	0.684375952995710	0.630491796612926	0.424368227377152	0.402504726858723	0.385549027181979	0.184621952955049
5	0.736658060056519	0.684375952995710	0.630491796612926	0.424368227377152	0.402504726858723	0.385549027181979	0.184621952955049	0.0194360298452843
6	0.684375952995710	0.630491796612926	0.424368227377152	0.402504726858723	0.385549027181979	0.184621952955049	0.0194360298452843	0.0156138817167138
7	0.630491796612926	0.424368227377152	0.402504726858723	0.385549027181979	0.184621952955049	0.0194360298452843	0.0156138817167138	0.122329070689410
8	0.424368227377152	0.402504726858723	0.385549027181979	0.184621952955049	0.0194360298452843	0.0156138817167138	0.122329070689410	0.254953544635778
9	0.402504726858723	0.385549027181979	0.184621952955049	0.0194360298452843	0.0156138817167138	0.122329070689410	0.254953544635778	0.690649968487609
10	0.385549027181979	0.184621952955049	0.0194360298452843	0.0156138817167138	0.122329070689410	0.254953544635778	0.690649968487609	0.843161811047635
11	0.184621952955049	0.0194360298452843	0.0156138817167138	0.122329070689410	0.254953544635778	0.690649968487609	0.843161811047635	0.365226584260069
12	0.0194360298452843	0.0156138817167138	0.122329070689410	0.254953544635778	0.690649968487609	0.843161811047635	0.365226584260069	0.349035314209039
13	0.0156138817167138	0.122329070689410	0.254953544635778	0.690649968487609	0.843161811047635	0.365226584260069	0.349035314209039	0.210584097424116
14	0.122329070689410	0.254953544635778	0.690649968487609	0.843161811047635	0.365226584260069	0.349035314209039	0.210584097424116	0.254579462052982
15	0.254953544635778	0.690649968487609	0.843161811047635	0.365226584260069	0.349035314209039	0.210584097424116	0.254579462052982	0.253034338341432
16	0.690649968487609	0.843161811047635	0.365226584260069	0.349035314209039	0.210584097424116	0.254579462052982	0.253034338341432	0.0870555228007400
17	0.843161811047635	0.365226584260069	0.349035314209039	0.210584097424116	0.254579462052982	0.253034338341432	0.0870555228007400	0.00461504055949743
18	0.365226584260069	0.349035314209039	0.210584097424116	0.254579462052982	0.253034338341432	0.0870555228007400	0.00461504055949743	0
19	0.349035314209039	0.210584097424116	0.254579462052982	0.253034338341432	0.0870555228007400	0.00461504055949743	0	0.104438164555675
20	0.210584097424116	0.254579462052982	0.253034338341432	0.0870555228007400	0.00461504055949743	0	0.104438164555675	0.302661272287393
21	0.254579462052982	0.253034338341432	0.0870555228007400	0.00461504055949743	0	0.104438164555675	0.302661272287393	0.538048671396914
22	0.253034338341432	0.0870555228007400	0.00461504055949743	0	0.104438164555675	0.302661272287393	0.538048671396914	0.675751723016244
23	0.0870555228007400	0.00461504055949743	0	0.104438164555675	0.302661272287393	0.538048671396914	0.675751723016244	0.432081647589810
24	0.00461504055949743	0	0.104438164555675	0.302661272287393	0.538048671396914	0.675751723016244	0.432081647589810	0.154467643889646
25	0	0.104438164555675	0.302661272287393	0.538048671396914	0.675751723016244	0.432081647589810	0.154467643889646	0.0940777034582308
26	0.104438164555675	0.302661272287393	0.538048671396914	0.675751723016244	0.432081647589810	0.154467643889646	0.0940777034582308	0.194441620753451
27	0.302661272287393	0.538048671396914	0.675751723016244	0.432081647589810	0.154467643889646	0.0940777034582308	0.194441620753451	0.292150364933824
28	0.538048671396914	0.675751723016244	0.432081647589810	0.154467643889646	0.0940777034582308	0.194441620753451	0.292150364933824	0.134242787728465
29	0.675751723016244	0.432081647589810	0.154467643889646	0.0940777034582308	0.194441620753451	0.292150364933824	0.134242787728465	0.0616423038607762
30	0.432081647589810	0.154467643889646	0.0940777034582308	0.194441620753451	0.292150364933824	0.134242787728465	0.0616423038607762	0.0674771789293919
31	0.154467643889646	0.0940777034582308	0.194441620753451	0.292150364933824	0.134242787728465	0.0616423038607762	0.0674771789293919	0.210978510582064
32	0.0940777034582308	0.194441620753451	0.292150364933824	0.134242787728465	0.0616423038607762	0.0674771789293919	0.210978510582064	0.766869294732348
33	0.194441620753451	0.292150364933824	0.134242787728465	0.0616423038607762	0.0674771789293919	0.210978510582064	0.766869294732348	0.885083457010999
34	0.292150364933824	0.134242787728465	0.0616423038607762	0.0674771789293919	0.210978510582064	0.766869294732348	0.885083457010999	0.828962937361498
35	0.134242787728465	0.0616423038607762	0.0674771789293919	0.210978510582064	0.766869294732348	0.885083457010999	0.828962937361498	0.458440238274341
36	0.0616423038607762	0.0674771789293919	0.210978510582064	0.766869294732348	0.885083457010999	0.828962937361498	0.458440238274341	0.178378433325879
37	0.0674771789293919	0.210978510582064	0.766869294732348	0.885083457010999	0.828962937361498	0.458440238274341	0.178378433325879	0.315064956187611
38	0.210978510582064	0.766869294732348	0.885083457010999	0.828962937361498	0.458440238274341	0.178378433325879	0.315064956187611	0.407699188810051
39	0.766869294732348	0.885083457010999	0.828962937361498	0.458440238274341	0.178378433325879	0.315064956187611	0.407699188810051	0.236881696383191
40	0.885083457010999	0.828962937361498	0.458440238274341	0.178378433325879	0.315064956187611	0.407699188810051	0.236881696383191	0.0535812308130197
41	0.828962937361498	0.458440238274341	0.178378433325879	0.315064956187611	0.407699188810051	0.236881696383191	0.0535812308130197	0.0340435480919755
42	0.458440238274341	0.178378433325879	0.315064956187611	0.407699188810051	0.236881696383191	0.0535812308130197	0.0340435480919755	0.052686685506333
43	0.178378433325879	0.315064956187611	0.407699188810051	0.236881696383191	0.0535812308130197	0.0340435480919755	0.052686685506333	0.457193974017525
44	0.315064956187611	0.407699188810051	0.236881696383191	0.0535812308130197	0.0340435480919755	0.052686685506333	0.457193974017525	0.989631406672495
45	0.407699188810051	0.236881696383191	0.0535812308130197	0.0340435480919755	0.052686685506333	0.457193974017525	0.989631406672495	1
46	0.236881696383191	0.0535812308130197	0.0340435480919755	0.052686685506333	0.457193974017525	0.989631406672495	1	0.902494561571147
47	0.0535812308130197	0.0340435480919755	0.052686685506333	0.457193974017525	0.989631406672495	1	0.902494561571147	0.280338300770529
48	0.0340435480919755	0.052686685506333	0.457193974017525	0.989631406672495	1	0.902494561571147	0.280338300770529	0.226004838676886
49	0.052686685506333	0.457193974017525	0.989631406672495	1	0.902494561571147	0.280338300770529	0.226004838676886	0.211702279057475
50	0.457193974017525	0.989631406672495	1	0.902494561571147	0.280338300770529	0.226004838676886	0.211702279057475	0.181735011283469