

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTA DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

**CLASIFICACIÓN DE ACCIDENTES DE TRÁNSITO  
EN LA ZONA DE VALPARAÍSO UTILIZANDO REDES  
NEURONALES CON ALGORITMOS PSO**

**JOSÉ DAVID FIERRO MOYA**

INFORME FINAL DEL PROYECTO  
PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO CIVIL EN INFORMÁTICA

JUNIO 2012

Pontificia Universidad Católica de Valparaíso  
Facultad de Ingeniería  
Escuela de Ingeniería Informática

**CLASIFICACIÓN DE ACCIDENTES DE TRÁNSITO  
EN LA ZONA DE VALPARAÍSO UTILIZANDO REDES  
NEURONALES CON ALGORITMOS PSO**

**JOSÉ DAVID FIERRO MOYA**

Profesor Guía: **Nibaldo Rodríguez Agurto**

Carrera: **Ingeniería Civil Informática**

JUNIO 2012

*Dedicado a Dios, a mis padres quienes me mostraron quien quiero ser, a mi hermana para quien quiero estar, a mi hermano en quien confío, a mi amada quien me enseña que la vida es mejor cuando se ama y a ti lector por leer lo que me apasiona.*

# Índice

Índice.....	i
Glosario de Términos.....	iv
Lista de Abreviaturas o Siglas.....	v
Lista de Símbolos.....	vi
Lista de Figuras.....	vii
Lista de Tablas.....	viii
Resumen.....	xi
Abstract.....	xii
1. Introducción.....	1
2. Marco General del Proyecto.....	3
2.1 Objetivo General.....	3
2.2 Objetivos Específicos.....	3
3. Marco Teórico.....	4
3.1 Redes Neuronales Artificiales.....	4
3.1.1 Reseña Histórica.....	5
3.1.2 Estado del Arte de Redes Neuronales en Accidentes de Tránsito.....	6
3.1.3 Fundamentos Biológicos.....	7
3.1.4 La Neurona Biológica.....	8
3.1.5 La Neurona Artificial.....	9
3.1.6 Clasificación de Redes Neuronales.....	11
3.1.7 Estructura de una Red.....	11
3.1.8 Aprendizaje.....	13
3.1.9 Comparación Entre Redes Neuronales Biológicas y Artificiales.....	16
3.1.10 PERCEPTRON.....	17
3.1.11 PERCEPTRON Multicapa (MLP).....	19
3.1.12 Red Neuronal de Base Radial (RBFN).....	22
3.1.13 Red Wavelet (WNN).....	26
3.1.14 Red Neuronal Polinomial (PNN).....	26
3.2 Optimización por Enjambre de Partículas.....	30
3.2.1 VARIANTES DE PSO.....	32
3.3 Método de Entrenamiento.....	37

4.	Descripción de Datos en Estudio .....	40
5.	Desarrollo de los Modelos .....	44
5.1	Tratamiento de Datos .....	45
5.1.1	Selección de Datos de Entrada y Codificación. ....	45
5.1.2	Normalización de los datos .....	46
5.1.3	Comparación de modelos .....	46
5.1.4	Entorno de trabajo .....	47
6.	Primeros Modelos propuestos.....	48
6.1	MLP.....	48
6.1.1	Elegir las funciones de activación a utilizar.....	48
6.1.2	Determinar la Cantidad de Nodos Ocultos.....	48
6.1.3	Entrenar red neuronal mediante algoritmo LDWPSO y evaluar que red presenta mejores resultados. ....	50
7.	Segundos Modelos propuestos.....	52
7.1	MLP.....	52
7.1.1	Elegir las funciones de activación a utilizar.....	52
7.1.2	Determinar la Cantidad de Nodos Ocultos.....	52
7.1.3	Entrenar y evaluar la red neuronal t-s mediante algoritmos LDWPSO, QPSO y LDWQPSO. ....	54
7.1.4	Entrenar y evaluar la red neuronal s-s mediante algoritmos LDWPSO, QPSO y LDWQPSO. ....	56
7.1.5	Comparación de resultados. ....	58
8.	Terceros Modelos propuestos .....	60
8.1	MLP.....	60
8.1.1	Elegir las funciones de activación a utilizar.....	60
8.1.2	Determinar la Cantidad de Nodos Ocultos.....	60
8.1.3	Entrenar y evaluar la red neuronal sigmoidal-softmax mediante algoritmos LDWPSO, QPSO y LDWQPSO.....	61
8.1.4	Comparación de resultados. ....	64
9.	Comparación de resultados.....	65
10.	Conclusión .....	68
11.	Bibliografía .....	70
	Anexos .....	73
A.	Plantilla SIEC 2 .....	74
B.	Resultados en la elección de nodos ocultos .....	76

C.	Resultados en el entrenamiento de la red .....	83
D.	Resultados en la evaluación de la red .....	88

## Glosario de Términos

Algoritmo Evolutivo: son algoritmo de optimización y búsqueda que se basan en el postulado de la evolución biológica.

*Feed forward*: hace referencia a las redes neuronales artificiales de propagación hacia adelante.

Heurística: es un algoritmo que permite encontrar buenas soluciones y/o en buenos tiempos.

Hiperplano: espacio vectorial de n dimensiones. En MLP es usado para separar clases.

Metaheurística: consiste en una estrategia de alto nivel que guía a otras heurísticas para buscar soluciones que sean factibles en dominios donde la tarea es compleja.

Principio de incertidumbre: establece que no se puede determinar simultáneamente y con precisión la posición y la velocidad, ya que entre más certeza se tenga de una, menos se tendrá de la otra.

Retro Propagación: *Back Propagation* en inglés, es un algoritmo de aprendizaje supervisado utilizado para entrenar redes neuronales artificiales.

## Lista de Abreviaturas o Siglas

BFGS	: Broydon Fletcher Goldfarb Shanno.
BP	: <i>Back Propagation</i> .
CONASET	: Comisión Nacional de Seguridad de Tránsito.
LDWPSO	: <i>Linearly Decreasing Weight Particle Swarm Optimization</i> .
LS-SVM	: <i>Least Squares Support Vector Machines</i> .
MLP	: <i>Multi Layer Perceptron</i> .
PNN	: <i>Polynomial Neural Network</i> .
PSO	: <i>Particle Swarm Optimization</i> .
RBFN	: <i>Radial Basis Function Network</i> .
SVM	: <i>Support Vector Machines</i> .
WNN	: <i>Wavelet Neural Network</i> .

## Lista de Símbolos

- $\epsilon$ : Umbral.
- $\phi$ : Función de base radial.
- $\Psi$ : Función Wavelet madre.
- $\omega$ : Coeficiente de inercia.
- $\lambda$ : Coeficiente de constricción.
- $\mu$ : Media.
- $\sigma$ : Desviación estándar.
- $\varphi$ : Valor aleatorio en el intervalo  $[0,1]$ .

## Lista de Figuras

Figura 3.1 Neurona biológica .....	8
Figura 3.2 Neurona artificial .....	10
Figura 3.3 Red neuronal artificial de tres capas .....	12
Figura 3.4 Aprendizaje supervisado .....	14
Figura 3.5 Aprendizaje no supervisado .....	15
Figura 3.6 Arquitectura de un PERCEPTRON con dos entradas y una salida .....	17
Figura 3.7 Separación de dos clases mediante un PERCEPTRON .....	18
Figura 3.8 Arquitectura del PERCEPTRON multicapa .....	20
Figura 3.9 Funciones de activación del PERCEPTRON multicapa .....	22
Figura 3.10 Arquitectura de una RBFN .....	23
Figura 3.11 Comparación MLP-RBFN .....	24
Figura 3.12 Curva RBFN .....	25
Figura 3.13 Arquitectura de una WNN .....	26
Figura 3.14 Arquitectura de una Pi-Sigma .....	28
Figura 3.15 Arquitectura de una Sigma-Pi .....	29
Figura 3.16 Uso de función <i>Softmax</i> .....	30
Figura 3.17 Actualización de una partícula [34] .....	32
Figura 3.18 División de Datos Para Validación Cruzada .....	39
Figura 3.19 División de Datos de entrenamiento Para Validación Cruzada .....	39
Figura 4.1 Entidades base de datos CONASET .....	40
Figura 5.1 Pasos para desarrollo del modelo .....	44
Figura 6.1 Error Promedio según el número de nodos ocultos para cada red propuesta, primeros modelos .....	50
Figura 7.1 Error de entropía cruzada promedio según el número de nodos ocultos para cada red propuesta, primeros modelos .....	53
Figura 7.2 Convergencias de tipos de PSO .....	55
Figura 7.3 Convergencias de tipos de PSO .....	58
Figura 8.1 Error de entropía cruzada promedio según el número de nodos ocultos para cada red propuesta, segundos modelos .....	61
Figura 8.2 Convergencias de tipos de PSO .....	63

## Lista de Tablas

Tabla 3.1 Algunas redes según el tipo de aprendizaje .....	15
Tabla 3.2 Diferencias entre redes neuronales artificiales y biológicas .....	16
Tabla 3.3 Matriz de Confusión .....	38
Tabla 5.1 Atributos de entrada del modelo .....	45
Tabla 5.2 Formación de clases [4] .....	46
Tabla 6.1 Mejores resultados en elección de nodos ocultos, primeros modelos .....	49
Tabla 6.2 Resultados promedio del entrenamiento, primeros modelos .....	51
Tabla 6.3 Resultados promedio de la evaluación .....	51
Tabla 6.4 Ganancia de los resultados promedio de la evaluación .....	51
Tabla 7.1 Mejores resultados en elección de nodos ocultos, segundos modelos .....	53
Tabla 7.2 Resultados promedios del entrenamiento de tablas C.5, C.6 y C.7, segundos modelos .....	54
Tabla 7.3 Resultados promedios de la evaluación en tablas D.5, D.6 y D.7, segundos modelos .....	55
Tabla 7.4 Resultados promedios del entrenamiento de tablas C.5, C.6 y C.7, segundos modelos .....	56
Tabla 7.5 Resultados promedios de la evaluación en tablas D.8, D.9 y D.10, segundos modelos .....	57
Tabla 7.6 Resultados promedio de la evaluación .....	59
Tabla 7.7 Ganancia de los resultados promedio de la evaluación .....	59
Tabla 8.1 Mejores resultados en elección de nodos ocultos, terceros modelos .....	61
Tabla 8.2 Resultados promedios del entrenamiento de tablas C.11, C.12 y C.13, terceros modelos .....	62
Tabla 8.3 Resultados promedios de la evaluación en tablas D.11, D.12 y D.13, terceros modelos .....	62
Tabla 8.4 Ganancia de los resultados promedio de la evaluación .....	64
Tabla 9.1 Comparación de modelos con una muestra de 3000 datos .....	65
Tabla 9.2 Resultados obtenidos de [4] contrastados con segundos y terceros modelos .....	66
Tabla 9.3 LS-SVM PSO vs Redes Bayesianas vs RNA QPSO .....	66
Tabla 9.4 Ganancia de los resultados promedio de la evaluación de los modelos de la tabla 9.3 .....	67
Tabla B.1 Resultados MLP con LDWPSO utilizando función tangente hiperbólica para capa oculta y de salida correspondientemente, primeros modelos .....	76
Tabla B.2 Resultados MLP con LDWPSO utilizando función sigmooidal y tangente hiperbólica para capa oculta y de salida correspondientemente, primeros modelos .....	77
Tabla B.3 Resultados MLP con LDWPSO utilizando función sigmooidal para capa oculta y de salida, primeros modelos .....	78
Tabla B.4 Resultados MLP con LDWPSO utilizando función tangente hiperbólica y sigmooidal para capa oculta y de salida correspondientemente, primeros modelos .....	79
Tabla B.5 Resultados de algoritmos PSO para elección de nodos ocultos, utilizando función de activación tangente hiperbólica y sigmooidal para capa oculta y de salida correspondientemente, segundos modelos propuestos .....	80
Tabla B.6 Resultados de algoritmos PSO para elección de nodos ocultos, utilizando función de activación sigmooidal para capa oculta y de salida, segundos modelos propuestos .....	80

Tabla B.7 Resultados de algoritmos PSO para elección de nodos ocultos, utilizando función de activación sigmoïdal para la capa oculta y softmax para la capa de salida, terceros modelos..	81
Tabla C.1 Resultados del entrenamiento para MLP con LWDPSO con 4 nodos ocultos y función tangente hiperbólica para capa oculta y de salida, primeros modelos .....	83
Tabla C.2 Resultados del entrenamiento para MLP con LWDPSO con 4 nodos ocultos y función de sigmoïdal y tangente hiperbólica para capa oculta y de salida correspondientemente, primeros modelos.....	83
Tabla C.3 Resultados del entrenamiento para MLP con LWDPSO con 5 nodos ocultos con función de activación sigmoïdal para capa oculta y de salida, primeros modelos .....	84
Tabla C.4 Resultados del entrenamiento para MLP con LWDPSO con 4 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, primeros modelos.....	84
Tabla C.5 Resultados del entrenamiento para MLP con LWDPSO con 5 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos.....	85
Tabla C.6 Resultados del entrenamiento para MLP con QPSO con 9 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos.....	85
Tabla C.7 Resultados del entrenamiento para MLP con QPSO con 9 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos.....	85
Tabla C.8 Resultados del entrenamiento para MLP con LWDPSO con 9 nodos ocultos con función sigmoïdal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos .....	86
Tabla C.9 Resultados del entrenamiento para MLP con QPSO con 5 nodos ocultos con función sigmoïdal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos .....	86
Tabla C.10 Resultados del entrenamiento para MLP con LDWQPSO con 4 nodos ocultos con función sigmoïdal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos .....	86
Tabla C.11 Resultados del entrenamiento para MLP con LWDPSO con 44 nodos ocultos con función sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos .....	87
Tabla C.12 Resultados del entrenamiento para MLP con QPSO con 50 nodos ocultos con función sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos .....	87
Tabla C.13 Resultados del entrenamiento para MLP con LWDQPSO con 27 nodos ocultos con función sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos .....	87
Tabla D.1 Resultados de la evaluación para MLP con LWDPSO con 4 nodos ocultos y función tangente hiperbólica para capa oculta y de salida, primeros modelos .....	88
Tabla D.2 Resultados de la evaluación para MLP con LWDPSO con 4 nodos ocultos y función de sigmoïdal y tangente hiperbólica para capa oculta y de salida correspondientemente, primeros modelos.....	88
Tabla D.3 Resultados de la evaluación para MLP con LWDPSO con 5 nodos ocultos con función de activación sigmoïdal para capa oculta y de salida, primeros modelos .....	89

Tabla D.4 Resultados de la evaluación para MLP con LWDPSO con 4 nodos ocultos con función de activación tangente hiperbólica y sigmoideal para la capa oculta y de salida correspondientemente, primeros modelos .....	89
Tabla D.5 Resultados de la evaluación para MLP con LWDPSO con 5 nodos ocultos con función de activación tangente hiperbólica y sigmoideal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos .....	90
Tabla D.6 Resultados de la evaluación para MLP con QPSO con 9 nodos ocultos con función de activación tangente hiperbólica y sigmoideal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos .....	90
Tabla D.7 Resultados de la evaluación para MLP con LDWQPSO con 9 nodos ocultos con función de activación tangente hiperbólica y sigmoideal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos .....	90
Tabla D.8 Resultados de la evaluación para MLP con LWDPSO con 9 nodos ocultos con función de activación sigmoideal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos .....	90
Tabla D.9 Resultados de la evaluación para MLP con QPSO con 5 nodos ocultos con función de activación sigmoideal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos .....	91
Tabla D.10 Resultados de la evaluación para MLP con LDWQPSO con 4 nodos ocultos con función de activación sigmoideal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos .....	91
Tabla D.11 Resultados de la evaluación para MLP con LWDPSO con 44 nodos ocultos con función de activación sigmoideal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos .....	91
Tabla C.12 Resultados de la evaluación para MLP con QPSO con 50 nodos ocultos con función de activación sigmoideal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos .....	91
Tabla D.13 Resultados de la evaluación para MLP con LWDQPSO con 27 nodos ocultos con función de activación sigmoideal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos .....	92

## Resumen

Si bien las redes neuronales artificiales fueron inventadas hace ya tiempo, su utilización en problemas como el desarrollado en el presente trabajo es relativamente reciente. Lo mismo ocurre con PSO, que ha mostrado tener buenos resultados en estudios anteriores, como por ejemplo [1]. Por ello, para abordar el problema de clasificación de accidentes de tránsito en Valparaíso se explican diversas técnicas de redes neuronales artificiales y PSO.

Mediante el marco teórico se abordan cuatro tipos de redes neuronales útiles para realizar clasificación: MLP (Perceptrón multicapa), RBFN (Red de función de base radial), WNN (red Wavelet) y PNN (Red neuronal polinomial). Así también se trata el algoritmo Retro Propagación, PSO y algunas de sus variantes, los cuales sirven para el entrenamiento de los pesos de la red neuronal artificial.

Con la entrega en esta etapa final se pretende introducir técnicas de clasificación basadas en redes neuronales artificiales y sus algoritmos de entrenamientos LDWPSO, QPSO y LDWQPSO, siendo utilizados con diferentes redes y tamaños de muestras. Además se mostrarán los resultados obtenidos del modelo MLP con diferentes funciones de activación, habiendo entrenado la red con LDWPSO, QPSO y LDWQPSO.

**Palabras Claves:** Red Neuronal Artificial, Optimización por Enjambre de Partículas, Accidentes de tránsito.

## Abstract

While artificial neural networks were invented long ago, its use in problems such as the one developed in this report is relatively recent. The same happens with PSO, which has shown good results in previous studies, for example [1]. Therefore, to address the problem of classification of traffic accidents in Valparaíso, this work explains various techniques of artificial neural networks and PSO.

Using the theoretical framework, four types of neural networks useful for classification are addressed here: MLP (Multi Layer Perceptron) RBFN (Radial Basis Function Network), WNN (Wavelet Neural Network) and PNN (Polynomial Neuronal Network). So too is treated the back propagation algorithm, PSO and some of its variants, which are used for training the weights of the artificial neural network.

With the delivery at this final stage, it is pretended to introduce classification techniques based on artificial neural networks and their training algorithms LDWPSO, QPSO and LDWQPSO, being used for different networks and sample sizes. It will also show the results of the MLP model with different activation functions, having trained the network with LDWPSO, QPSO and LDWQPSO.

**Keywords:** Artificial Neural Network, Optimization by Particle Swarm, Traffic Accidents.

# 1. Introducción

En Chile los costos por siniestros de tránsito en el año 2009 llegaron a un total de 357.462.613.856 pesos chilenos, esto considerando los costos sociales y privados. Estos costos consideran un total de 56330 siniestros de tránsito de los cuales 6662 pertenecen a la región de Valparaíso, es decir, aproximadamente un 11.8% de los siniestros de tránsito [2].

Una de las interrogantes que surge es como lograr una aceptable clasificación de los accidentes en estos siniestros, específicamente en la región de Valparaíso, siendo la segunda región en donde ocurre la mayor cantidad de ellos, para así poder conocer mejor las consecuencias de determinadas condiciones de tránsito para tomar conciencia y reducir los costos tanto sociales como privados. Con lo anterior en mente es que se recurrirá al estudio de redes neuronales artificiales y algoritmos evolutivos para lograr así una clasificación en cuanto al tipo de accidente, es decir, determinar si una persona resulta ilesa o lesionada en un accidente de tránsito dado los atributos de entrada de la red.

Actualmente si bien hay investigaciones en accidentes de tránsito mediante diversas técnicas (redes Bayesianas, MLP y SVM entre otras), el uso de redes neuronales artificiales en clasificación de accidentes de tránsito es reducido y su enfoque dependerá de cómo el investigador desee construir el modelo, es por ello que esta investigación se centra en el uso de redes neuronales artificiales para la clasificación de accidentes de tránsito en Valparaíso.

Existe una gran cantidad de redes neuronales, es por ello que abordarlas todas en el tiempo programado para esta investigación es imposible, por lo mismo en el presente trabajo se han descrito cuatro técnicas de redes neuronales útiles para la clasificación, dentro de las que se encuentran MLP (Perceptrón multicapa), RBFN (Red de función de base radial), WNN (red Wavelet) y PNN (Red neuronal polinomial), cada una de ellas plantea una diferencia fundamental consistente en la función de activación usada por las neuronas, la cual describe la forma en que se logra la clasificación.

Las redes neuronales artificiales requieren entrenar sus pesos para lograr una buena clasificación, es por ello que se ha seleccionado el algoritmo evolutivo PSO para dicha tarea. Si bien existen otros métodos como *Back Propagation* (del cual se da una breve introducción), se eligió PSO por su simplicidad y eficiencia [3]. En este estudio también se presentan variaciones de PSO.

En el presente trabajo se ha utilizado la red neuronal MLP para determinar el tipo de estado con que resultan las personas involucradas en un accidente de tránsito en la región de Valparaíso, para ello se proponen diversas redes neuronales en donde sus principales diferencias radican en cinco aspectos: número de nodos ocultos, número de nodos de salida, el valor de los pesos sinápticos, las funciones de activación de los nodos ocultos y las funciones de activaciones de los nodos de salida.

El número de nodos ocultos se determinó comparando la cantidad de nodos que ofrecen los mejores resultados. El número de nodos de salida fue de uno o dos. Las funciones de

activación para los nodos de la capa oculta fueron la función sigmoïdal o la función tangente hiperbólica. Las funciones de activación para los nodos de la capa de salida fueron la función sigmoïdal o la función tangente hiperbólica en caso de una neurona de salida, en caso de dos neuronas de salida fue la función Softmax. Los pesos sinápticos fueron determinados mediante el entrenamiento de la red, para ello se realizó un entrenamiento mediante variantes de PSO, entre ellas se encuentran: LDWPSO, QPSO, LDWQPSO. La cantidad de partículas utilizadas para las variantes de PSO fue de 100. En cuanto a la función de desempeño utilizada por estas variantes de PSO, estas fueron: exactitud o entropía cruzada.

Los resultados obtenidos son comparados entre las diferentes técnicas utilizadas mediante tablas comparativas que facilitan su estudio, así también los resultados presentes son comparados con los resultados presentes en dos trabajos en estudio [4] LS-SVM basado en Optimización por Enjambres de Partículas para Clasificación de Accidentes de Tránsito y [5] Estructuras Bayesianas en Accidentes de Tránsito.

En cuanto a [4] LS-SVM basado en Optimización por Enjambres de Partículas para Clasificación de Accidentes de Tránsito, este aborda el problema de clasificación mediante una técnica de SVM, LS-SVM (*Least Squares Support Vector Machines*), utilizando las siguientes variantes de PSO para su entrenamiento: PSO F.I.L, IPSO y DAPSO. El mejor resultado es obtenido con PSO F.I.L. En cuanto a [5] Estructuras Bayesianas en Accidentes de Tránsito, este utiliza estructuras Bayesianas, mas los resultados obtenidos por esta técnica, como se verá, pueden ser superados tanto por LS-SVM como por MLP expuestos en este trabajo.

En el capítulo nueve se pueden ver con mayor detalle la comparación de los resultados obtenidos.

Es así que el presente trabajo tiene por objetivo un estudio respecto a algunas técnicas de clasificación con la intención de proponer modelos de clasificación y comparar sus resultados.

## **2. Marco General del Proyecto**

### **2.1 Objetivo General**

Desarrollar un modelo de clasificación del grado de severidad de las lesiones con que resultan las personas en accidentes de tránsito en la zona de Valparaíso de Chile utilizando redes neuronales con algoritmos de optimización por enjambre de partículas.

### **2.2 Objetivos Específicos**

- Explicar el funcionamiento de redes neuronales artificiales y PSO.
- Implementar y estimar los parámetros de un clasificador neuronal usando algoritmos PSO.
- Evaluar y contrastar los modelos de clasificación propuestos.

## 3. Marco Teórico

### 3.1 Redes Neuronales Artificiales

Una de las cosas que diferencia al ser humano del resto de los seres vivos es su capacidad de aprender y su inteligencia, las que le permite crear cosas de tal complejidad que están lejos del alcance de cualquier otra especie que habite este planeta. Es así que con su inteligencia el ser humano ha desarrollado a lo largo de su historia objetos que facilitan su vida, por lo que desde la invención de la rueda, el ser humano no ha parado y ha llegado a crear inventos con la capacidad de realizar diversas tareas, ya sea de forma mecánica y/o lógica. Pero hay un límite en las invenciones mencionadas, estas realizan solo tareas para las que fueron creadas sin la capacidad de aprender, siguiendo alguna secuencia de pasos ya establecidos, de modo que surge la siguiente pregunta, ¿Cómo superar ese límite?, es aquí que el ser humano mira hacia él mismo y encuentra en su naturaleza la respuesta. La manera de superar el límite planteado es su capacidad de aprender y actuar acorde a lo aprendido.

Así entonces la solución a la limitante anterior consiste construir sistemas inteligentes, este objetivo es precisamente el objetivo de la inteligencia artificial.

En la inteligencia artificial se pueden apreciar dos áreas significativas:

- Inteligencia artificial simbólica.
- Inteligencia artificial sub simbólica.

La primera de ellas se encarga de construir sistemas que contenga características que se pueden determinar cómo inteligentes, para lo cual es preciso definir el problema y diseñar un sistema capaz de resolverlo siguiendo esquemas prefijados por la disciplina. Un ejemplo son los sistemas expertos.

En la segunda de ellas no se hace un diseño de alto nivel de sistemas que resuelvan los problemas utilizando las técnicas de la disciplina, sino que se usa un sistema genérico que se va adaptando y construyéndose para formar por sí mismo un sistema capaz de resolver el problema. Es en la inteligencia artificial sub simbólica donde se estudian los mecanismos del sistema nervioso, cerebro, sus características, estructura y funcionamiento con el fin de diseñar programas que recojan esto y se adapten generando sistemas que puedan resolver problemas. Es en este campo en donde se encuentran las redes neuronales artificiales.

Las redes neuronales artificiales tienen como objetivo emular el comportamiento de una red neural biológica, de manera de poder resolver problemas, para lo cual toma ciertos elementos de la red neuronal biológica como lo son las neuronas y crea un elemento análogo a este llamado neurona artificial, es por ello que para conocer el funcionamiento de una red neuronal artificial, es preciso adentrarse en los fundamentos biológicos que son la base para la construcción de una red neuronal artificial. Teniendo claro lo anterior es que posteriormente se ahondará más en profundidad sobre los fundamentos biológicos de las redes neuronales así como los principales elementos que son considerados a la hora de emularle en una red neuronal artificial.

### 3.1.1 Reseña Histórica

La primera implementación de redes neuronales artificiales fue un dispositivo hidráulico descrito por Russell en 1913, pero no fue sino hasta la década de los 40 en que las redes neuronales artificiales tomaron fuerza y evolucionaron hasta nuestros días.

Warren McCulloch y Walter Pitts [6] fueron los primeros en modelar matemáticamente redes neuronales artificiales, en el modelo se plantea la idea de que las neuronas operan mediante impulsos binarios. También se introduce la idea de una función de paso por umbral. Mas el factor clave del modelo es la capacidad de aprendizaje.

Donald Hebb en 1949 desarrollo un procedimiento matemático de aprendizaje conocido como aprendizaje Hebbiano. Este trabajo se describe en detalle en el libro “*Organization of Behavior*” [7].

Marvin Minsky en 1951 obtuvo los primeros resultados prácticos de redes neuronales artificiales [8], posteriormente Minsky y Edmons diseñaron una maquina con cuarenta neuronas con conexiones que se ajustaban según los sucesos que ocurrían al realizar una serie de tareas. Esta máquina estaba construida con tubos, motores y relés, logrando modelar con éxito el comportamiento de una rata buscando comida en un laberinto.

Frank Rosenblatt en 1957 creó el modelo llamado PERCEPTRON, el cual tomo como base el modelo de McCulloch, al cual se le añadió aprendizaje [9]. Rosenblatt desarrollo un modelo de dos niveles, el cual tomaba el error entre las salidas deseadas y las salidas ocurridas y según este ajustaba los pesos de sus conexiones. Posteriormente intento con tres niveles, pero no encontró un buen modelo matemático para entrenar las conexiones con la capa oculta. Algunos de sus trabajos relacionados a PERCEPTRON están en su libro “*Principles of Neurodynamics*” [10].

Bernard Widrow en 1959 [11] basándose en el modelo PERCEPTRON creó una red neuronal llamada ADALINE (*Adaptative Linear Element*), el cual al igual que el PERCEPTRON ajusta sus pesos dependiendo del error entre la salida deseada y la salida ocurrida. La mayor diferencia entre ADALINE y PERCEPTRON está en su función de transferencia, mientras que PERCEPTRON usa una función de transferencia escalón, ADALINE usa una función de transferencia lineal. Tanto ADALINE como PERCEPTRON tienen el mismo problema, solo pueden resolver problemas linealmente separables.

Steinbuch a principios de los años sesenta desarrollo métodos de codificación de información en redes neuronales. Sus redes se aplicaron en control de múltiples procesos de producción, diagnostico de fallos en maquinarias y reconocimiento de escritura a mano distorsionada.

Marvin Minsky y Seymour Paper en 1969 frenaron el crecimiento de las investigaciones sobre redes neuronales. Publicaron el libro “*Perceptrons*” [12] en el cual probaron matemáticamente que el PERCEPTRON no era capaz de resolver problemas relativamente

fáciles, tales como el aprendizaje de una función no lineal (extensamente empleadas en la computación y en los problemas del mundo real).

Paul Werbos en 1974 desarrollo la idea básica del algoritmo de aprendizaje de propagación hacia atrás (*Brack Propagation*), cuyo significado quedo definitivamente aclarado en 1985.

Stephen Grossberg en 1977 plantea la teoría de resonancia adaptada (TRA), la cual es una arquitectura de red que se diferencia de las demás previamente inventadas. Esta red puede simular memoria a corto y a largo plazo.

John Hopfield en 1985 causa el resurgimiento de las redes neuronales con su libro “Computación neuronal de decisiones en problemas de optimización”.

David Rumelhart y G. Hinton en 1986 redescubren el algoritmo de aprendizaje de propagación hacia atrás (*Back Propagation*).

Desde 1986 en adelante el panorama para las redes neuronales artificiales ha sido alentador, cada vez son más las investigaciones y trabajos que se realizan con respecto a las redes neuronales, y de apoco diversas ramas de investigación han visto su potencial.

### **3.1.2 Estado del Arte de Redes Neuronales en Accidentes de Tránsito**

El estudio [13] realizado en 1997 por Abdelwahab y Abdel-Aty sobre los datos de accidentes en el área de Florida, tenía por objetivo analizar los accidentes ocurridos en intersecciones señalizadas. Las lesiones se clasificaron en: no lesiones, posibles lesiones y lesiones incapacitantes. En el estudio se comparó el desempeño de *Fuzzy ARTMAP* y el Perceptrón Multicapa, dando mejores resultados este último. Para MLP se utilizó Lavenberg-Marquardt como algoritmo de entrenamiento, obteniendo una precisión del 60.4% al probarlo. Para el caso de *Fuzzy ARTMAP* se logró una precisión del 56.1%.

Mussone realizó un estudio [14] para analizar los accidentes vehiculares ocurridos en las intersecciones en Milán, Italia. Se usó un modelo MLP *feed-forward* y se entrenó con BP. El modelo tenía diez nodos de entrada para ocho variables (día o noche, flujo de tránsito circulante en la intersección, número de puntos virtuales conflictivos, número de puntos real de real conflicto, tipos de intersección, tipos de accidentes, condiciones de la superficie del camino y condiciones del tiempo). Los nodos de salida fueron llamados índice de accidente, este fue calculado como un ratio entre el número de accidentes para una cierta intersección y el número de accidentes para la intersección más peligrosa. Los resultados mostraron que el mayor índice de accidentes ocurrió en intersecciones no señalizadas en la noche.

El año 2002 So Young Sohn y Sung Ho Lee [15] usaron varios algoritmos para mejorar la exactitud de clasificadores individuales (red neuronal y árbol de decisión) para dos categorías de gravedad en accidentes de tránsito (lesiones corporales y daño de propiedad). Se usaron tres enfoques distintos *classifier fusión* basado en el algoritmo de Dempster-Shafer, procedimiento Bayesiano y modelo logístico; *data ensemble fusión* de datos basado en *arcing*

y *bagging*; y *clustering* basado en el algoritmo k-means. Obteniendo los mejores resultados del algoritmo *clustering* basado en clasificación, con un 73.94% y un 76.1% para el método de *clustering* para red neuronal y árbol de decisión respectivamente. La red neuronal obtuvo un 70.86%. Los demás resultados se pueden ver en [15].

En [16] se realizó una clasificación de accidentes según la severidad del accidente, clasificando este en cinco clases (no lesión, posible lesión, lesiones no incapacitantes, lesiones incapacitantes y lesiones fatales) y se compararon las técnicas de árbol de decisión, red neuronal (usando un entrenamiento que combinaba *Back Propagation* y *conjugate gradient descent*), *hibrid* DT-ANN y SVM usando la función de base radial para el *kernel*. Como resultado se obtuvo que en todas las clases el árbol de decisión presentó mejor precisión. Según el orden de las clases mencionadas anteriormente, la red neuronal obtuvo como resultado 60.45%, 57.58%, 56.8%, 61.32% y 75.51%, con respecto al árbol de decisión obtuvo como resultado 67.54%, 64.40%, 60.37%, 71.38% y 89.46%, para el híbrido DT-ANN se obtuvo como resultado 65.12%, 63.10%, 62.24%, 72.63 y 90%, para SVM se obtuvo como resultado 62.88%, 3.67%, 3.21%, 2.83% y 3.33%.

En el año 2008 Jua Zúñiga presentó un modelo [17] de redes bayesianas para identificar los factores determinantes en los accidentes de tránsito que afecten a las personas, en este estudio se obtuvieron buenos resultados.

Se realizó un estudio [18] que comparó dos modelos analíticos (una modificación del modelo Smeed y el modelo de Andreassen adaptado) y una red neuronal artificial para estimar el número de muertos en accidentes de tránsito. Para probarlo se utilizaron los datos históricos de 20 años de tres grandes metrópolis de Turquía. Los resultados mostraron que la red neuronal artificial predecía mejor los resultados al obtener menores errores.

En Julio del 2010 Félix Castro presentó un modelo de clasificación de accidentes de tránsito [4], para lo cual usó un modelo LS-SVM basado en Optimización por enjambre de partículas. Este modelo se basa en el hecho de seleccionar los parámetros que usará LS-SVM mediante un algoritmo de PSO, una vez se obtengan estos parámetros se vuelve a entrenar para obtener una buena clasificación. Para este trabajo se clasificó en lesionados e ilesos, dando como mejor resultado una exactitud del 70%.

### **3.1.3 Fundamentos Biológicos**

Las redes neuronales artificiales se basan en el sistema de comunicación neuronal, el cual está formado por el sistema hormonal y por el sistema nervioso, estos sistemas se conectan con diversos órganos, lo que permitiendo obtener información del medio, procesar y almacenar información para posteriormente enviarla como órdenes a los diferentes órganos efectores. Este sistema está formado de tres partes:

Receptores: estos se ubican en las células sensoriales y como su nombre lo indica son entes receptores encargados de recibir los estímulos del medio (externo o proveniente del interior del organismo).

Sistema nervioso: es el encargado de recibir la información obtenida por los receptores, procesarla, almacenarla y enviarla a los órganos efectores como a otras partes del sistema nervioso.

Órganos efectores: son los que reciben la información enviada por el sistema nervioso para así realizar la acción.

### 3.1.4 La Neurona Biológica

La célula nerviosa llamada neurona es la unidad estructural y funcional más esencial del sistema de comunicación neuronal. Cada una de ellas debe desempeñar cuatro funciones:

- Recibir información del entorno, ya sea interna o externa al cuerpo, o de otras neuronas.
- Integra la información recibida y producir una señal apropiada.
- Conducir la señal a su terminal nerviosa.
- Transmitir la señal a otras células nerviosas, glándulas o músculos.

Aun cuando existe variación entre las neuronas, se pueden definir en todas ellas cuatro regiones bien definidas que realizan las funciones ya mencionadas. Dichas regiones son las dendritas, el cuerpo celular, el axón y las terminales sinápticas.

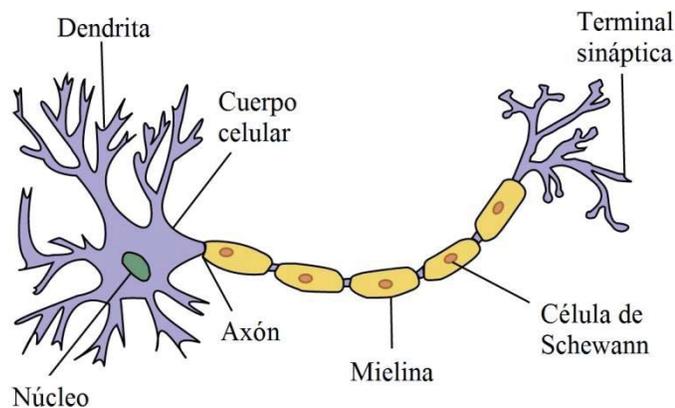


Figura 3.1 Neurona biológica

Dendritas: son prolongaciones del cuerpo de la célula nerviosa, ellas responden a señales de otras neuronas o del entorno exterior, encargándose de recibir información. En las neuronas del cerebro y la medula espinal, responden a los neurotransmisores químicos liberados por otras neuronas. Estas dendritas tienen en su membrana receptores proteicos que se unen a neurotransmisores específicos y, como resultado producen señales eléctricas. Las dendritas de las neuronas sensoriales tienen ciertas adaptaciones en su membrana permitiéndoles producir señales eléctricas en respuesta a estímulos específicos del entorno, como presión, luz, calor o moléculas odoríferas.

Cuerpo celular: encargado de cumplir la segunda función, que es la de integrar la información recibida y producir una señal apropiada. A él viajan las señales recibidas por las dendritas, las que pueden ser positivas o negativas, las que se suman. Si la suma tiene una magnitud positiva suficiente, la neurona producirá un potencial de acción, el cual es la señal eléctrica de salida de la neurona, además en su interior alberga al núcleo, el cual se encarga de realizar actividades metabólicas de la neurona.

Axón: Son líneas de distribución que llevan potenciales de acción desde el cuerpo celular a las terminales sinápticas. La mielina es un aislante que cubre el axón, funcionando como un aislante que acelera la conducción de la señal eléctrica.

Terminales sinápticas: son engrosamientos en los extremos ramificados de los axones, son las terminales las encargadas de transmitir la señal a otras células. El punto en que las terminales sinápticas se comunican con otras células se llama sinapsis.

Una neurona inactiva tiene un potencial eléctrico constante, este potencial es llamado potencial de reposo, siempre es negativo dentro de la célula, variando entre -40 y -90 minivolts. Si la célula recibe un estímulo, su potencial se puede hacer más o menos negativo. Si esta magnitud negativa se reduce lo suficiente como para alcanzar un nivel llamado umbral, ocurrirá un potencial de acción, en el cual el potencial de la neurona se elevará rápidamente hasta cerca de los 50 minivolts dentro de la célula. El potencial de acción dura unos pocos milisegundos, volviendo posteriormente la célula a su potencial normal. La carga positiva del potencial de acción fluye por el axón hasta las terminales sinápticas, donde la señal se comunica a otra célula a través de una sinapsis.

Cuando un potencial de acción llega a una terminal sináptica, el interior de la terminal adquiere una carga positiva. Esta carga tiene como efecto la liberación de neurotransmisores químicos al espacio entre las células, estos neurotransmisores se difunden rápidamente al otro lado de ese espacio y se unen a receptores en la membrana de la neurona post sináptica antes de alejarse por difusión y ser reabsorbidos por la neurona pre sináptica, así se transmite la señal entre neuronas.

Si la suma que se obtiene de los potenciales eléctricos mediante las dendritas alcanza un valor entre los 10 a 15 mini volts se alcanzara el umbral.

En el cerebro existen alrededor de 100000 millones de neuronas, donde cada una de ellas se conecta con alrededor de 10000 neuronas, es decir que cada actividad neuronal afecta a otras 10000 neuronas, con lo cual se forma una red de un gran tamaño. En la actualidad un dispositivo así está lejos del alcance. Los intentos más cercanos son de un millón de procesadores, donde cada procesador se conecta solamente con ocho adyacentes. Los sistemas artificiales actualmente existentes se ven bajo esta limitante.

### **3.1.5 La Neurona Artificial**

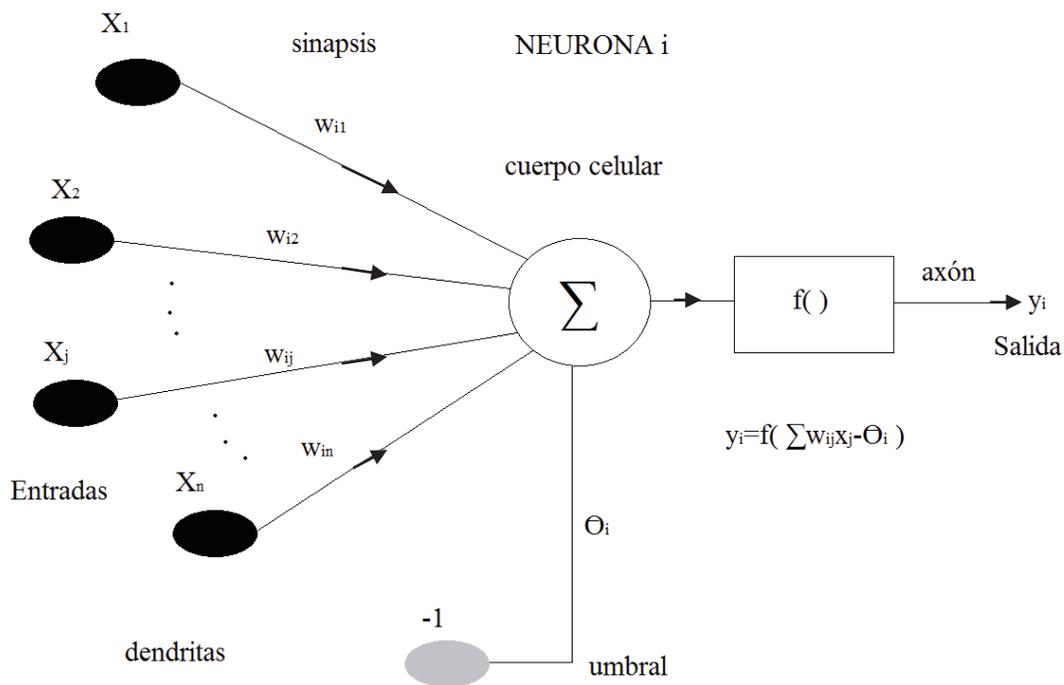
La neurona artificial es una emulación a menor escala de la neurona biológica. Posee un estado interno, el cual se conoce como nivel de activación, este estado puede

cambiar según las señales que reciba, las que pueden provenir del medio o de otras neuronas artificiales.

El conjunto de estados que puede poseer la neurona se denomina S, el cual puede tener valores binarios, naturales o incluso valores en un intervalo continuo.

La neurona tiene un nivel de activación, el cual está determinado por las entradas y los pesos sinápticos que esta posea, si bien las entradas son externas a ella, los pesos pueden cambiar para así adaptar el nivel de activación. Este cambio va a depender de las entradas que recibe la neurona mediante una función de activación, estas entradas pueden venir de otras neuronas o del exterior.

Como se desea calcular el nivel de activación, es preciso en primer lugar calcular la entrada total de la célula  $E_i$ , este valor se calcula mediante la fórmula 3.1.1, en donde se suma cada entrada ( $X=[x_1, x_2, \dots, x_j, \dots, x_n]$ , representando las señales de la sinapsis en una neurona biológica) multiplicada por un peso sináptico ( $W=[w_{i1}, w_{i2}, \dots, w_{ij}, \dots, w_{in}]$ , representando la fuerza de una conexión sináptica) y se le resta un umbral  $\Theta$ .



**Figura 3.2 Neurona artificial**

$$E = (X^T \times W) - \theta = x_1 \times w_1 + \dots + x_n \times w_n - \theta = \sum_{i=1}^n w_i \times x_i - \theta \quad (3.1.1)$$

La entrada total de la célula, E, es procesada por la función de activación  $f()$ , produciendo una señal de salida S. Es así que dependiendo de la función  $f()$  es que se tendrán diferentes modelos de autómatas.

$$S = f(E) \quad (3.1.2)$$

### 3.1.6 Clasificación de Redes Neuronales

Las neuronas artificiales se unen para formar una estructura neuronal o modelo conexionista, los que pueden ser clasificados según [19]:

El número de capas:

- Redes neuronales mono capa: esta clase de redes son las más simples, si bien estas redes tienen dos capas, la primera capa se encarga de recoger las entradas sin realizar cálculos sobre ellas por lo que no se considera como una capa para efectos de cálculo, es en la segunda capa (capa de salida) donde se realizan todos los cálculos.
- Redes neuronales multicapa: es una generalización de la anterior, en este tipo de redes existen un número de capas entre la capa de entrada y la capa de salida llamadas capas ocultas.

El tipo de conexiones:

- Redes neuronales no recurrentes: Esta estructura no posee memoria pues las señales se propagan solamente hacia adelante, sin existir la posibilidad de realimentación. Estas redes también se conocen como redes de propagación hacia adelante, en inglés *Feed-Forward*.
- Redes neuronales recurrentes: esta estructura es adecuada para el estudio de la dinámica de sistemas no lineales. En ella se permite la realimentación, ya sea entre neuronas de la misma capa, de diferentes capas o de la neurona misma. Estas redes también se conocen como redes de retro propagación, en inglés *Feed-Back*.

El grado de conexión:

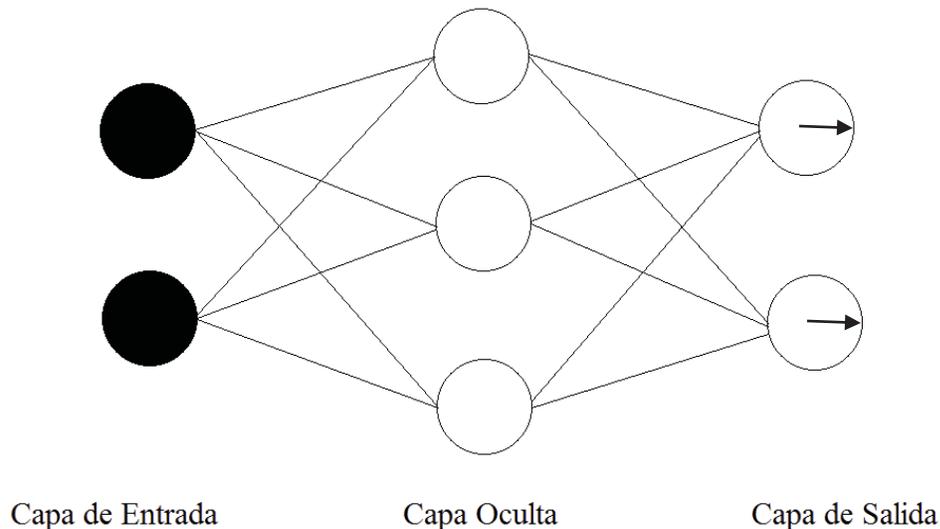
- Redes neuronales totalmente conectadas: Todas las neuronas de una capa se encuentran totalmente conectadas con las de la capa siguiente o anterior.
- Redes parcialmente conectadas: En este caso no se da la conexión completa entre las neuronas de diferentes capas.

### 3.1.7 Estructura de una Red

Anteriormente se entró en detalle respecto a la neurona artificial, pero este elemento por sí solo no es mucho lo que puede hacer, más al conectarse estas unidades forman un elemento más complejo llamado red neuronal artificial. Estas neuronas se organizan en capas, en donde cada capa cumple una función, es así que tenemos tres tipos de capas dentro de una red neuronal, la capa de entrada, la capa oculta y la capa de salida. Como se puede observar en la figura 3.3, la red parte con la capa de entrada, esta capa cumple la función de recibir los datos que vienen del exterior de la red neuronal propagando los datos a la siguiente capa de la red sin modificar dichos datos, por lo que no se considera al momento de contabilizar el total de capas de la red neuronal, después viene la capa oculta, la cual se encuentra entre la capa de

entrada y la capa de salida de la red, esta capa puede albergar más de una capa oculta, esto puede favorecer a resolver problemas más complejos, pero aumentará la complejidad de la red, al final de la red neuronal se encuentra la capa de salida, la cual arroja los resultados de salida de la red neuronal.

Como se aprecia en la figura 3.3 se representa una red neuronal de tres capas, estas capas se conectan mediante conexiones, las cuales tendrán pesos sinápticos. Para conocer cuántos pesos sinápticos posee la red basta con multiplicar la cantidad de neuronas de cada capa por la cantidad de neuronas de la siguiente capa, por ejemplo si tomamos la figura 3.3 obtendremos que la red tiene  $2 \times 3 + 3 \times 2 = 12$  pesos sinápticos.



**Figura 3.3 Red neuronal artificial de tres capas**

Como se pudo ver, las neuronas artificiales al conectarse ofrecen un potente elemento, la red neuronal artificial, estas redes se caracterizan principalmente por [20]:

- La naturaleza de lo que emulan, tienen una inclinación a la adquisición de conocimiento por medio de las experiencias, este conocimiento es almacenado al igual que en el cerebro en el peso relativo de las conexiones inter neuronales.
- Poseer gran plasticidad y adaptabilidad, teniendo la capacidad de cambiar dinámicamente junto con el medio.
- Tener tolerancia a fallas, del mismo modo como ocurre con los sistemas biológicos, pueden sufrir daños considerables y aun así continuar teniendo un buen comportamiento.
- Tener un comportamiento no-lineal, esto les permite obtener y procesar la información proveniente de fenómenos no lineales.

Es así como se estructura a grandes rasgos una red neuronal artificial, si bien puede parecer simple respecto a la estructura neuronal biológica, este tipo de emulación ha logrado buenos resultados en diversas áreas.

### 3.1.8 Aprendizaje

El aprendizaje o entrenamiento es la fase más importante dentro de las redes neuronales artificiales, esto debido a que el tipo de problemas que podrá resolver la red estarán determinados por el esquema de aprendizaje ocupado. Esto se debe a que la red neuronal artificial es una emulación de la red neuronal biológica, y como tal su aprendizaje estará basado en los ejemplos con los que aprenda. Desde este matiz se puede decir que el conjunto de ejemplos con los que aprenda una red neuronal debe ser significativo y representativo. Significativo en cuanto a que la cantidad de ejemplos con los que se entrene la red debe ser lo suficientemente amplio como para que la red pueda adaptar sus pesos de manera eficaz, de lo contrario los resultados no serán satisfactorios. Representativo, en el sentido que los componentes del conjunto de aprendizaje deben ser diversos, de tal manera que la red no se especialice en un tipo de ejemplo por sobre el resto, puesto que implicaría que su aplicación no sea general.

Con el aprendizaje se pretende poder resolver problemas de manera eficaz, esto mediante la calibración de los pesos de todas las conexiones de la red.

Para calibrar los pesos se van introduciendo los ejemplos del conjunto de aprendizaje, con cada ejemplo los pesos de irán adaptando conforme el esquema de aprendizaje utilizado, si después de ingresados todos los ejemplos la red aun no converge conforme cierto criterio, entonces se vuelve a entrenar con el conjunto de ejemplos. La modificación de los pesos puede hacerse durante o después de ingresados todos los ejemplos.

El periodo de aprendizaje puede ser determinado según [21]:

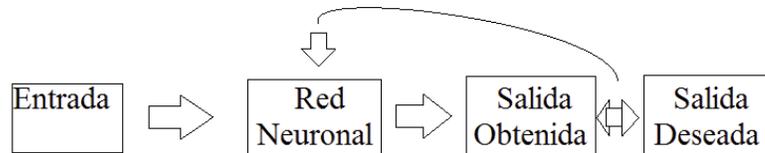
- Un número fijo de ciclos: después de un cierto número fijo de ciclos se acepta la red resultante.
- El error sea inferior a una cantidad preestablecida. De ser el caso, primero se debe definir una función de error a nivel de patrón individual y a nivel de la totalidad del conjunto de entrenamiento. Este valor se decide a priori. Este criterio debe ser acompañado de algún otro como por ejemplo el anterior, esto en caso de que la red no converja (señal de que la red no ha alcanzado una solución y por ende requerirá modificaciones).
- La modificación de los pesos sea irrelevante. Básicamente es cuando la variación de los pesos no es significativa o no se produce variación de los pesos en ninguna conexión.

Los tipos de aprendizaje que se pueden dar son: aprendizaje supervisado, no supervisado, por refuerzo e híbrido.

#### 3.1.8.1 Aprendizaje Supervisado

El aprendizaje supervisado como su nombre lo sugiere, es un aprendizaje en el cual existe un control de las salidas, este tipo de control se puede hacer mediante tres formas:

- **Aprendizaje por corrección:** en el cual los datos de aprendizaje tienen dos tipos de atributos, los datos propiamente tal y los datos relacionados a la solución del problema. La forma más habitual de modificar los pesos de las conexiones se puede observar en la figura 3.4. Cuando se introduce un ejemplo este es procesado, la salida obtenida es comparada con la salida esperada. La diferencia entre la salida esperada y la obtenida influirá en la modificación de los pesos, es decir, si la diferencia es leve los pesos variarán poco, de lo contrario la variación en los pesos será mayor.



**Figura 3.4 Aprendizaje supervisado**

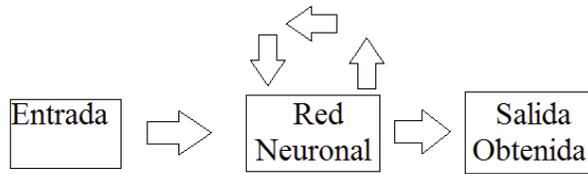
- **Aprendizaje por refuerzo:** en este caso solo se conoce si la salida de la red es o no correcta, en otras palabras la información es de tipo booleana (verdadero o falso)
- **Aprendizaje estocástico:** en este caso se hacen cambios aleatorios en los valores de los pesos de las conexiones de la red y se evalúa su efecto según el objetivo deseado y distribuciones de probabilidad. Se suele hacer una analogía con termodinámica, pues se considera un estado de energía  $E$  el que simboliza la estabilidad. Si después de realizar los cambios el estado de energía es menor, se aceptará el cambio, en caso contrario se aceptaría solo dependiendo de la distribución de probabilidad.

### 3.1.8.2 Aprendizaje No Supervisado

Este tipo de aprendizaje a diferencia del anterior no posee los datos relacionados con la solución del problema, por lo que no hay datos que permitan guiar el proceso de aprendizaje. Por lo cual los pesos de la red se modificarán a partir de la información interna, entonces la red tratará de determinar características de los datos del conjunto de entrenamiento.

Por lo general suelen haber dos tipos de aprendizaje no supervisado:

- **Aprendizaje Hebbiano:** en él los pesos de las conexiones se ajustan según la correlación que halla entre las neuronas. Si las dos neuronas son activas (positivas) se reforzará la conexión, pero si una es activa y la otra pasiva (negativa) se debilita la conexión.
- **Aprendizaje Competitivo o cooperativo:** En este caso las neuronas cooperan y compiten entre ellas para realizar una tarea. Al presentar información de entrada, solo un grupo o una neurona de salida se activará, anulando a las demás.



**Figura 3.5 Aprendizaje no supervisado**

### 3.1.8.3 Aprendizaje Híbrido

Como su nombre lo indica es una mezcla de aprendizajes, esta se realiza utilizando el aprendizaje supervisado y el aprendizaje no supervisado. Este aprendizaje puede ser empleado en redes neuronales artificiales de base radial [21], para ello se utilizan dos fases, en la primera (fase no supervisada) se determinan los centros y amplitudes de las neuronas en la capa oculta, para luego en la segunda fase (fase supervisada) determinar los pesos y umbrales de la capa de salida.

### 3.1.8.4 Ejemplos de Redes Según el Tipo de Aprendizaje

**Tabla 3.1 Algunas redes según el tipo de aprendizaje**

<ul style="list-style-type: none"> <li>• Aprendizaje Supervisado <ul style="list-style-type: none"> <li>○ Aprendizaje por corrección de error <ul style="list-style-type: none"> <li>▪ Perceptrón</li> <li>▪ Regla Delta (Madaline)</li> <li>▪ <i>Backpropagation</i></li> <li>▪ <i>Counterpropagation</i></li> </ul> </li> <li>○ Aprendizaje por refuerzo <ul style="list-style-type: none"> <li>▪ LRP (<i>Linear Reward Penalty</i>)</li> <li>▪ ARP (<i>Associative Reward Penalty</i>)</li> <li>▪ <i>Adaptive Heuristic Critic</i></li> </ul> </li> <li>○ Aprendizaje estocástico <ul style="list-style-type: none"> <li>▪ Máquina de Boltzmann</li> <li>▪ Máquina de Cauchy</li> </ul> </li> </ul> </li> <li>• Aprendizaje No-Supervisado <ul style="list-style-type: none"> <li>○ Aprendizaje Hebbiano <ul style="list-style-type: none"> <li>▪ Hopfield</li> <li>▪ <i>Linear Associative Memory</i></li> <li>▪ <i>Fuzzy Associative Memory</i></li> <li>▪ Grossberg</li> <li>▪ <i>Bidirectional Associative Memory</i></li> </ul> </li> <li>○ Aprendizaje competitivo/cooperativo <ul style="list-style-type: none"> <li>▪ <i>Learning vector quantizer</i></li> <li>▪ Cognitrón/Neocognitrón</li> <li>▪ Teoría de Resonancia Adaptiva</li> </ul> </li> </ul> </li> <li>• Aprendizaje Híbrido <ul style="list-style-type: none"> <li>○ <i>Radial basis network</i></li> </ul> </li> </ul>
---

Según el tipo de aprendizaje es posible clasificar algunas redes neuronales artificiales como se muestra en la siguiente tabla 3.1 obtenida de [22] en la cual se agregó el aprendizaje híbrido:

### 3.1.9 Comparación Entre Redes Neuronales Biológicas y Artificiales

**Tabla 3.2 Diferencias entre redes neuronales artificiales y biológicas**

Redes neuronales Artificiales	Redes neuronales Biológicas
La sinapsis es un proceso simple y de baja complejidad comparada con las redes neuronales biológicas	La sinapsis tiene miles de componentes y procesos activos que se encargan de propagar impulsos electro-químicos
Los parámetros se actualizan en intervalos de tiempo discretos de forma periódica.	La información se propaga por impulsos electro-químico, los cuales se reciben en cualquier momento, es decir hay un funcionamiento asíncrono.
Necesitan muchas más presentaciones de patrones (dependiendo del problema, cientos o miles) para lograr una generalización aceptable	Aprende a partir de pocas presentaciones de patrones mediante un corto periodo de entrenamiento
Tiene una arquitectura simple en relación a las interconexiones entre unidades de proceso (por capas bien definidas), las cuales son fijas	Está estructurada por niveles, formando una malla de conexiones muy densa, la que parece no tener estructura y varía con el tiempo
No existe una red que resuelva cualquier tipo de problemas	Tiene la capacidad de aprender de manera simultánea una gran cantidad de tareas muy diversas
Los valores que se transfieren de unidad en unidad pueden ser calculados de manera relativamente precisos, esto en parte gracias a su simplicidad	Los impulsos no se prestan a una medida precisa de los valores implicados en las reacciones electro-químicas

Como se ha visto anteriormente, las redes neuronales artificiales son una emulación de las redes neuronales biológicas, es por ello que toma las características más importantes de la red neuronal biológica y les trata de imitar. A continuación se presentara una tabla que muestra una comparación entre las redes neuronales biológicas y las redes neuronales artificiales [21] basada en la tabla presente en [23].

### 3.1.10 PERCEPTRON

Dentro de las redes neuronales artificiales, una de las más conocidas es el PERCEPTRON, la cual fue diseñada como un sistema capaz de realizar tareas de clasificación de clases de manera automática. La idea es disponer de un conjunto de ejemplos cuyo fin fuera el de entrenar la red, para que después esta pudiera determinar las ecuaciones de las superficies que hacían de frontera para separar las dichas clases. De esta forma el sistema podrá determinar a qué clase pertenecen los nuevos ejemplos.

Su arquitectura consta de una estructura mono capa en la cual hay un conjunto de células de entrada y una o más células de salida, donde cada célula de entrada está conectada con todas las células de salida

Como se puede apreciar en la figura 3.6, la red consta de entradas, en este casos  $x_1$  y  $x_2$ , una salida “ $y$ ” y un umbral  $\Theta$ , el cual es utilizado como un factor de comparación para producir la salida.

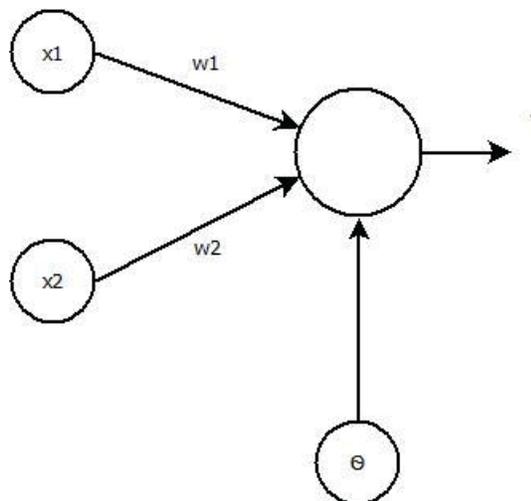


Figura 3.6 Arquitectura de un PERCEPTRON con dos entradas y una salida

Para obtener la salida de la red [21] primero se calcula la activación de la célula de salida mediante la suma ponderada por los pesos de todas las entradas:

$$y' = \sum_{i=1}^n w_i \times x_i \quad (3.1.3)$$

En el PERCEPTRON la función de salida es una función escalón que dependerá del umbral:

$$y = F(y', \Theta) \quad (3.1.4)$$

$$F(s, \Theta) = \begin{cases} 1 & \text{si } s > \Theta \\ -1 & \text{en otro caso} \end{cases} \quad (3.1.5)$$

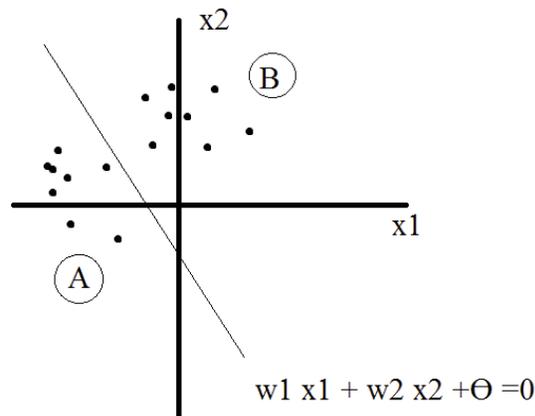
Según pase  $\Theta$  al lado contrario de la ecuación, es posible obtener las siguientes ecuaciones:

$$y = \sum_{i=1}^n w_i \times x_i + \Theta \quad (3.1.6)$$

$$F(s) = \begin{cases} 1 & \text{si } s > 0 \\ -1 & \text{en otro caso} \end{cases} \quad (3.1.7)$$

Con estas ecuaciones ya es posible determinar fácilmente la forma en que se puede clasificar mediante el modelo del PERCEPTRON.

Como se puede apreciar en la figura 3.7, si se produce una salida -1, la entrada pertenece a la clase B. Si se produce una salida 1, la entrada pertenece a la clase A.



**Figura 3.7 Separación de dos clases mediante un PERCEPTRON**

Como se aprecia en la figura este ejemplo es para dos dimensiones, expresado en  $n$  dimensiones se podría expresar de la siguiente forma [21]:

“Dados conjuntos de puntos en  $R^n$ :  $A=(a_1, \dots, a_{na})$  y  $B=(b_1, \dots, b_{nb})$ , obtener el hiperplano:  $w_1 a_1 + \dots + w_n a_n + \Theta = 0$ , de tal forma que”:

$$\forall \vec{a} \in A: w_1 \times a_1 + \dots + w_n \times a_n + \theta > 0 \quad (3.1.8)$$

$$\forall \vec{b} \in B: w_1 \times b_1 + \dots + w_n \times b_n + \theta < 0 \quad (3.1.9)$$

Entonces para la red se requerirían encontrar los valores que hagan:

$$\forall \vec{a} \in A: y(\vec{a}) = 1 \quad (3.1.10)$$

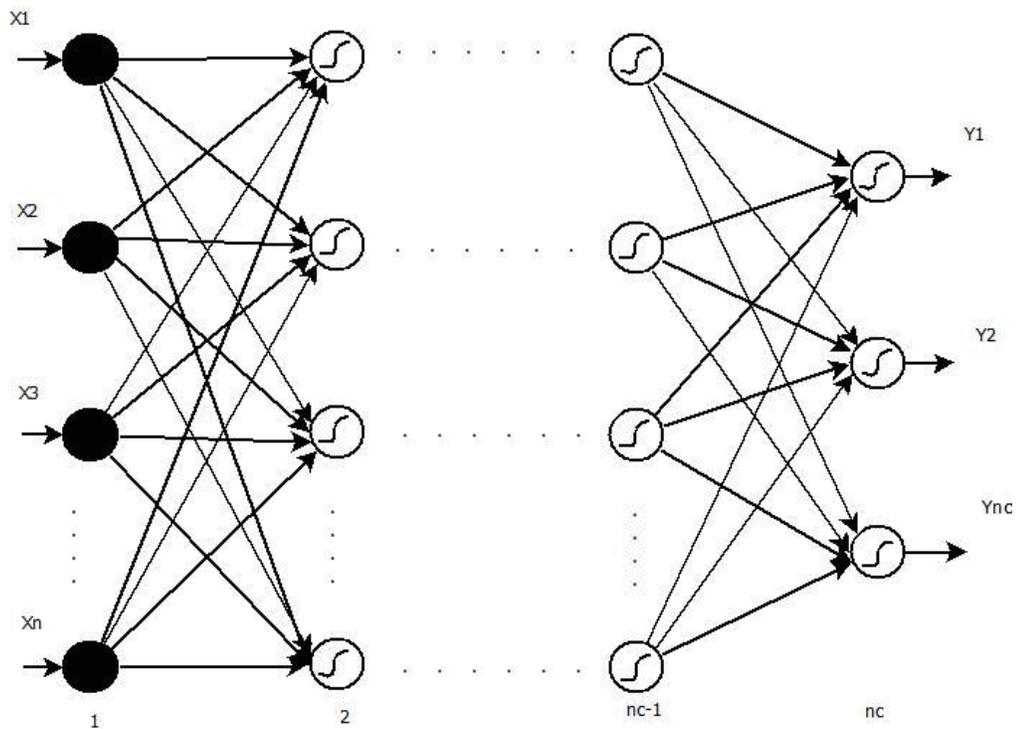
$$\forall \vec{b} \in B: y(\vec{b}) = -1 \quad (3.1.11)$$

Con lo cual ya no es una recta la que separa los puntos, sino un hiperplano.

Para el modelo del PERCEPTRON se utilizan iteraciones de modo que los pesos de las conexiones se modifiquen paulatinamente.

### 3.1.11 PERCEPTRON Multicapa (MLP)

El PERCEPTRON multicapa (MLP, según sus siglas en inglés *Multi Layer PERCEPTRON*) o red multicapa con conexiones hacia adelante, el cual se muestra en la figura 3.8, surge como consecuencia de las limitantes presentadas por el PERCEPTRON simple en lo relacionado al problema de separabilidad no lineal, estas limitantes se explican en [12].



**Figura 3.8 Arquitectura del PERCEPTRON multicapa**

El PERCEPTRON multicapa es una aproximación universal, dado que cualquier función continua puede ser aproximada mediante el PERCEPTRON multicapa, con al menos una capa oculta de neuronas, aun cuando no implica que sea necesariamente el mejor aproximador universal.

En la actualidad las redes de PERCEPTRON multicapa son unas de las más utilizadas para la resolución de problemas como por ejemplo reconocimiento del habla [24] y predicciones de series temporales [25] entre otros, esto por su capacidad de aproximador universal, su fácil uso y su aplicabilidad.

Aun cuando las redes de PERCEPTRON multicapa sean las más utilizadas, no son de las más potentes, además de tener limitaciones como por ejemplo el gran número de ejemplos que necesita para aprender en problemas complejos que requieren un gran número de variables.

Esta arquitectura se caracteriza por agrupar neuronas por capas, dentro de las cuales se encuentran:

- Capa de entrada: en esta capa se encuentran las neuronas encargadas de recibir las señales o patrones del exterior para luego propagarlos a la siguiente capa que es la primera capa oculta.
- Capas ocultas: las capas ocultas son las capas que se encuentran entre la capa de entrada y la capa de salida, pudiendo haber una o más capas ocultas. Las neuronas de las capas ocultas realizan un procesamiento no lineal de los patrones recibidos.

- Capa de salida: encargada de proporcionar al exterior la respuesta de la red para cada uno de los patrones de entrada.

Por lo general cada neurona de la capa  $t$  está conectada con todas las neuronas de la capa  $t+1$ , estas conexiones están dirigidas hacia adelante, razón por la cual se les conoce como redes *feed forward* (alimentadas hacia adelante), además estas conexiones tienen un determinado peso que se va ajustando en la etapa de entrenamiento. Todas las neuronas de la red tienen un umbral que se trata como una conexión más a la neurona, su valor es constante e igual a uno.

Ahora para entender más en profundidad el PERCEPTRON multicapa se debe considerar:

- $C$  capas, con  $C-2$  capas ocultas y  $n_c$  neuronas en la capa  $c$ , para  $c=1,2,\dots,C$ .
- $W^c=(w_{ij}^c)$  la matriz de pesos asociada a las conexiones de la capa  $c$  a las  $c+1$  para  $c=1,2,\dots,C-1$ , con  $w_{ij}^c$  es el peso de la conexión de la neurona  $i$  de la capa  $c$  a la neurona  $j$  de la capa  $c+1$ .
- $U^c=(u_i^c)$  es el vector de umbrales de las neuronas de la capa  $c$  para  $c=2,\dots,C$ .
- $a_i^c$  es la activación de la neurona  $i$  de la capa  $c$ . Esta se calcula de tres formas según sea el caso.

La activación de la capa de entrada para  $i=1, 2, 3, \dots, n_1$  se obtiene según la siguiente ecuación:

$$a_i^1 = x_i \quad (3.1.12)$$

La activación de las neuronas de la capa oculta para  $i=1, 2, 3, \dots, n_c$  y  $c=2,3,\dots,C-1$  se obtiene según la siguiente ecuación:

$$a_i^c = f\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \times a_j^{c-1} + u_i^c\right) \quad (3.1.13)$$

La activación de las neuronas de la capa de salida para  $i=1, 2, 3, \dots, n_c$  se obtiene según la siguiente ecuación:

$$y_i = a_i^c = f\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \times a_j^{c-1} + u_i^c\right) \quad (3.1.14)$$

Las funciones de activación más utilizadas se pueden apreciar en la figura 3.9, estas son:

- La función sigmoideal, cuya imagen está en el rango  $[0,1]$

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (3.1.15)$$

- La función tangente hiperbólica, cuya imagen está en el rango  $[-1,1]$

$$f_2(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (3.1.16)$$

Ambas funciones se relacionan mediante la siguiente ecuación

$$f_2(x) = 2 \times f_1(x) - 1 \quad (3.1.17)$$

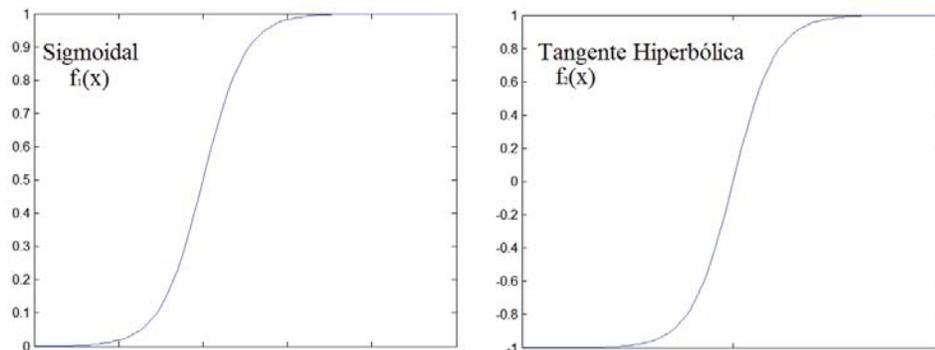


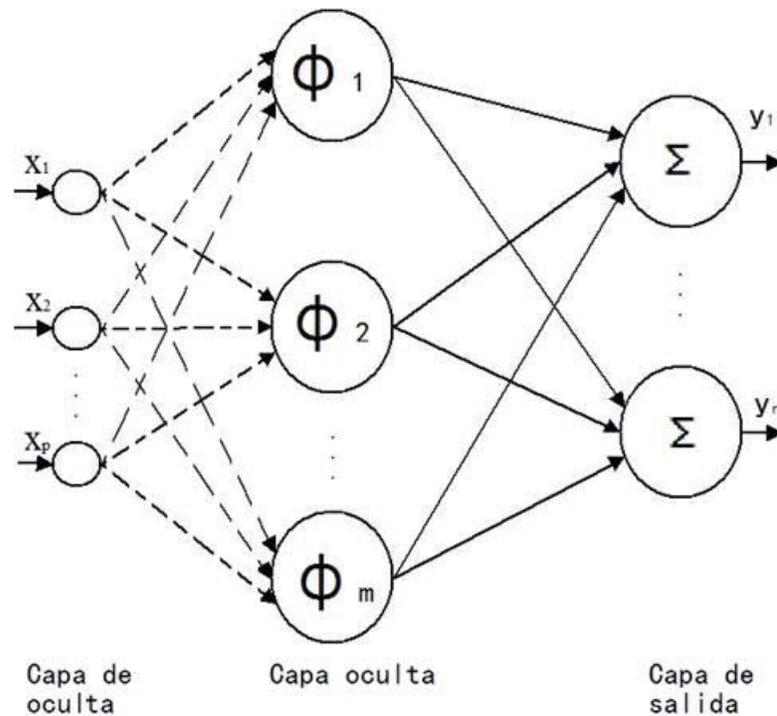
Figura 3.9 Funciones de activación del PERCEPTRON multicapa

La función de activación elegida dependerá de tipo de problema o de cómo se quieran expresar las salidas. Por lo general se suele usar una misma función para todas las neuronas de la red, pero esto puede variar según lo requiera el creador de la red.

### 3.1.12 Red Neuronal de Base Radial (RBFN)

La red neuronal de base radial (RBFN, según sus siglas en inglés *Radial Basis Function Network*), la cual se aprecia en la figura 3.10, es una red multicapa con conexiones hacia adelante, también es un aproximador universal al igual que el MLP, en el sentido que puede aproximar cualquier función continua en  $\mathbb{R}^n$ .

Esta fue pensada para aprender en un tiempo menor al que toma el MLP, a modo de poder usarse para aplicaciones en tiempo real. Esto se logra gracias a la arquitectura de la RBFN, como muestra la figura 3.10, tiene una sola capa oculta, en la cual se alojan neuronas con funciones de activación local, donde cada neurona de la capa oculta se activa en una región diferente del espacio de patrones de entrada, de modo que solo unas pocas neuronas sean procesadas para cada patrón de entrada.



**Figura 3.10** Arquitectura de una RBFN

En cuanto a las diferencias con respecto al MLP, se puede decir que:

Mientras el MLP usa aproximaciones globales y basadas en hiper planos, la RBFN usa aproximaciones locales dadas por las funciones de base radial, las cuales definen hiper esferas o hiper elipses, siendo la salida de la red resultado de una combinación lineal de las funciones de base radial, es decir de las salidas de las capas ocultas. Esta diferencia se puede ver en la siguiente figura.

Las conexiones entre la capa de entrada y la oculta no tienen pesos, pero las conexiones entre la capa oculta y la capa de salida si tienen pesos.

En las RBFN solo las neuronas de salida tienen umbral.

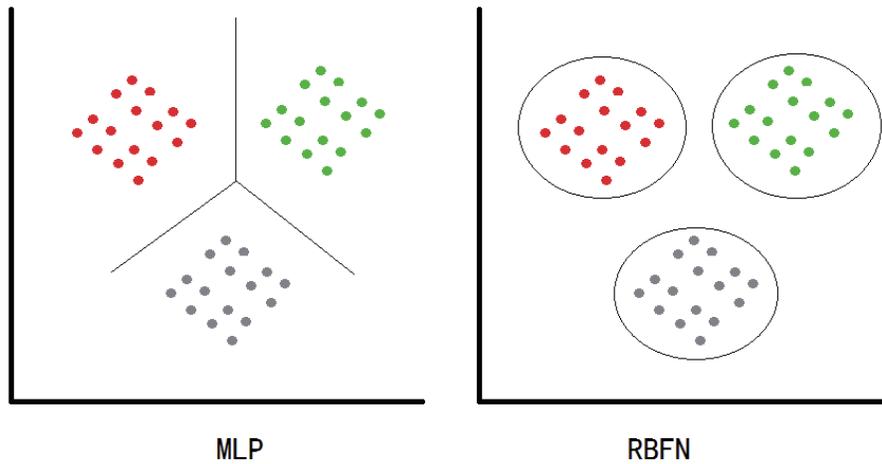


Figura 3.11 Comparación MLP-RBFN

Ahora que ya se entiende a grandes rasgos que es una RBFN, se explicará cómo funciona basándose en la explicación dada por [21].

Tomando una red RBFN con  $p$  neuronas de entrada,  $m$  neuronas en la capa oculta y  $r$  de salida, donde el vector de entrada está dado por  $X=(x_1, x_2, \dots, x_p)$  y el vector de salida por  $Y=(y_1, y_2, \dots, y_r)$ , este último se obtiene de la siguiente ecuación donde  $k=1, 2, \dots, r$ :

$$y_k = \sum_{i=1}^m w_{ik} \times \phi_i + u_k \quad (3.1.18)$$

Donde  $w_{ik}$  son los pesos entre la conexión de la neurona oculta  $i$  a la de salida,  $\phi_i$  es la activación de la neurona oculta y  $u_k$  es el umbral de la capa de salida  $k$ .

Se puede observar claramente entonces que la salida de la red es una combinación lineal de las activaciones de las neuronas de la capa oculta.

La función  $\phi_i$ , conocida como función de base radial, es la que determina la activación de la neurona de la capa oculta, la cual está definida por la siguiente ecuación, para  $i=1, 2, \dots, m$ :

$$\phi_i = \phi \left( \frac{\|X - C_i\|}{d_i} \right) \quad (3.1.19)$$

Donde  $\phi$  es la función de base radial,  $C_i=(c_{i1}, c_{i2}, \dots, c_{ip})$  son los vectores de los centros de la función de base radial,  $d_i$  es un número real que representa la desviación de la función de base radial y  $\|X-C_i\|$  representa la distancia euclidiana entre el vector  $X$  y el vector  $C_i$ , esta distancia está definida por la siguiente ecuación:

$$\|X - C_i\| = \left( \sum_{j=1}^p (x_j - c_{ij})^2 \right)^{\frac{1}{2}} \quad (3.1.20)$$

La función de base radial es de carácter local, esta se acerca al máximo cuando el patrón de entrada está cerca del centro, y se acerca al mínimo cuando el patrón de entrada se aleja del centro.

Las funciones de base radial (las que describen una curva como se muestra en la figura 3.12) más utilizadas, donde  $c, \Theta > 0$  y  $r$  pertenecen a los reales, son:

- Función gaussiana:

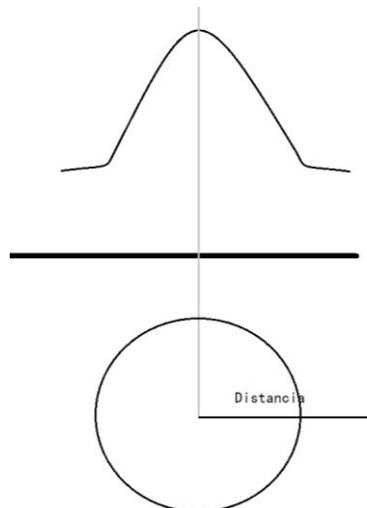
$$\phi(r) = e^{\left(\frac{-r^2}{2 \times \Theta^2}\right)} \quad (3.1.21)$$

- Función inversa cuadrática:

$$\phi(r) = \frac{1}{c^2 + r^2} \quad (3.1.22)$$

- Función inversa multicuadrática:

$$\phi(r) = \frac{1}{\sqrt{c^2 + r^2}} \quad (3.1.23)$$



**Figura 3.12 Curva RBFN**

### 3.1.13 Red Wavelet (WNN)

Las técnicas Wavelet pueden ofrecer un conocimiento y rendimiento agregado en el análisis de datos en situaciones donde las técnicas de Fourier han sido usadas.

Las redes neuronales Wavelet (WNN, según sus siglas en inglés *Wavelet Neural Network*) son útiles en tareas de clasificación, predicción y estimación de funciones [26].

Este tipo de redes tienen una estructura similar a las RBFN, con la diferencia de la función usada en la capa oculta, como se puede observar en la figura 3.13, basada en [27].

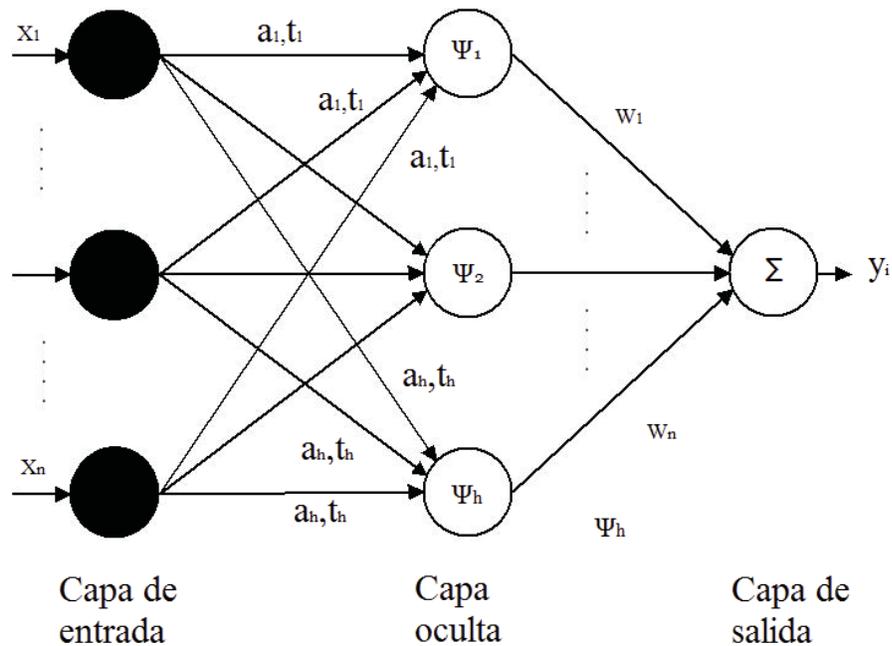


Figura 3.13 Arquitectura de una WNN

En donde se utiliza la ecuación basada en la transformada discreta en el tiempo de las Wavelets:

$$\psi\left(\frac{x - t_i}{a_i}\right) \times \frac{1}{\sqrt{a_i}} \quad (3.1.24)$$

Donde  $\psi$  es una función Wavelet madre, hay varias de estas funciones [27], más una de las más utilizadas es la Segunda Derivada de una Gaussiana:

$$\psi(t) = (1 - t^2) \times e^{-\frac{t^2}{2}} \quad (3.1.25)$$

### 3.1.14 Red Neuronal Polinomial (PNN)

Las redes neuronales polinomiales (PNN, según sus siglas en inglés *Polynomial Neural Network*) son del tipo *feed-forward*, estas no tienen un único enfoque, derecho existen una

gran cantidad de enfoques [28] (implementaciones de red neuronal de modelos discretos Volterra, implementadas, MLP con una capa oculta de funciones polinomiales de activación, Pi-Sigma y Sigma-Pi entre otras), esto se debe a su potencial dado que tienen habilidades de aproximación.

Las principales diferencias en cada enfoque radican según [28] en:

- La topología de la red polinomial y cuál es su conectividad.
- Cuales funciones de activación polinomial son usadas en los nodos de la red (lineales, cuadráticas o asignaciones altamente no lineales en una o varias variables)
- Cuál es la técnica de aprendizaje para los pesos.
- Si hay algoritmos diseñados que busquen la adecuada estructura de la red polinomial.
- Qué criterios son considerado para el control de la búsqueda al evaluar el desempeño de los datos.

Dado que la cantidad de redes PNN es amplia a continuación se darán a conocer algunas de ellas.

### 3.1.14.1 Red Pi-Sigma

La red Pi-Sigma [29] como se puede observar en la figura 3.14, tiene una sola capa oculta y dado que los pesos solo se ubican entre la capa de entrada y la capa oculta, el entrenamiento de la red resulta más rápido. Aun así esta red tiene la desventaja de no ser un aproximador universal [28].

La salida de la red está dada por la siguiente ecuación [30]:

$$y_i = \theta \left( \prod_{j=1}^K h_j \right) \quad (3.1.26)$$

Con:

$$h_j = \theta_j + \sum_{k=1}^N w_{kj} \times x_k \quad (3.1.27)$$

Donde  $X=(x_1, x_2, \dots, x_k, \dots, x_N)$ ,  $w_{kj}$  es el peso entre la k-esima y la j-esima neurona,  $h_j$  es la salida de la suma de la j-esima neurona,  $\theta_j$  es umbral ajustable de la suma de la j-esima unidad y  $\theta(r)$  es la función de activación no lineal, como por ejemplo:

$$\theta(r) = \frac{1}{(1 + e^{-r})} \quad (3.1.28)$$

$$\theta(r) = r \quad (3.1.29)$$

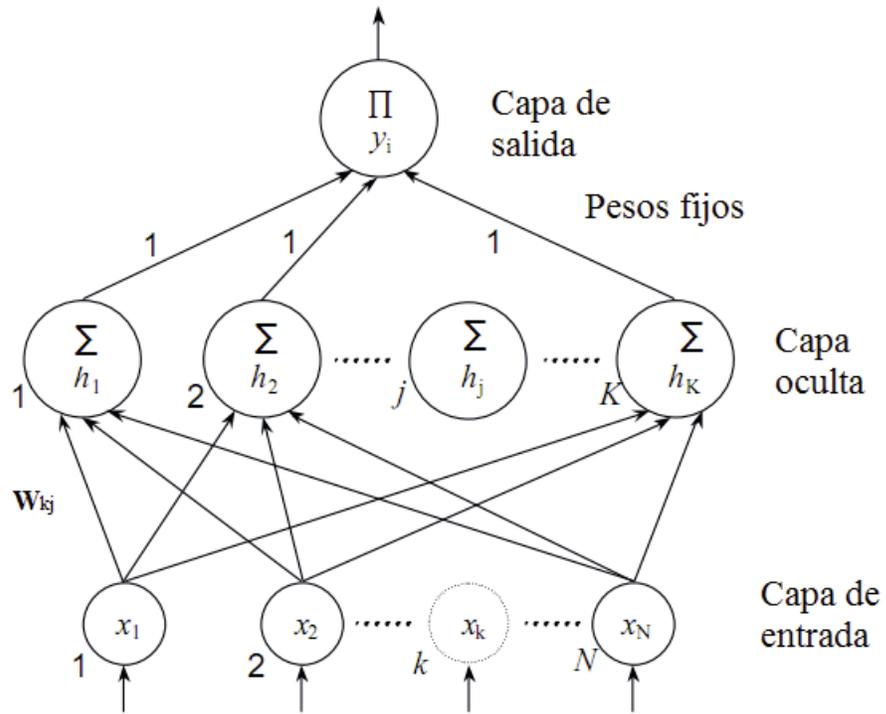


Figura 3.14 Arquitectura de una Pi-Sigma

### 3.1.14.2 Sigma-Pi

Este tipo de red como se puede ver en la figura 3.15 es muy similar a la red Pi-Sigma, teniendo también una capa de entrada, una capa oculta y una capa de salida, y al igual que Pi-sigma solo tiene pesos entre una de sus capas. La diferencia es el orden, ya que los pesos solo van entre la capa oculta y la capa de salida, además la función usada en la capa de oculta esta descrita por la siguiente ecuación.

$$h_j = \prod_{k=1}^N x_k \quad (3.1.30)$$

Mientras que la salida está dada por la siguiente ecuación.

$$y_i = \theta_i + \sum_{j=1}^K w_{ji} \times h_j \quad (3.1.31)$$

Donde  $X=(x_1, x_2, \dots, x_k, \dots, x_N)$ ,  $w_{ji}$  es el peso entre la  $j$ -ésima y la  $i$ -ésima neurona,  $h_j$  es la salida de la suma de la  $j$ -ésima neurona,  $\Theta_i$  es umbral ajustable de la suma de la  $i$ -ésima unidad y  $\theta(r)$  es la función de activación no lineal expresada anteriormente.

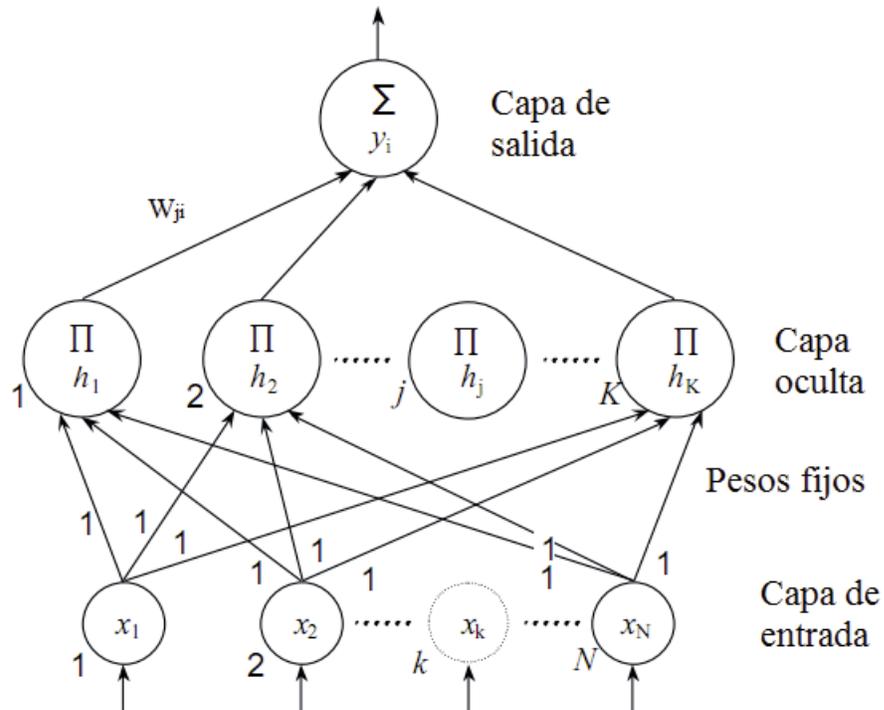


Figura 3.15 Arquitectura de una Sigma-Pi

Una adaptación de esta última red puede darse al considerar los pesos fijos como pesos variables.

### 3.1.14.3 Softmax

En caso de haber más de una neurona de salida, se necesita una función discriminante múltiple, en tal caso se utiliza una función logística múltiple denominada función de activación exponencial normalizada o *softmax*, en vez de una función umbral múltiple. La función *softmax* tiene la cualidad de permitir interpretar las múltiples salidas de la red como probabilidad, cumpliendo tres propiedades importantes:

1. Las salidas han de ser mayores que cero.
2. Las salidas deben estar entre cero y uno.
3. La suma de las salidas debe ser uno.

La función *softmax* está dada por la siguiente ecuación:

$$g_j(x, \theta_j) = \frac{e^{f_j(x, \theta_j)}}{\sum_{i=1}^J e^{f_j(x, \theta_j)}} \quad (3.1.32)$$

Donde  $g_j(x, \theta_j)$  es la probabilidad de que el patrón  $x$  pertenezca a la clase  $j$ . Su uso se puede apreciar en la figura 3.16 tomada de [31].

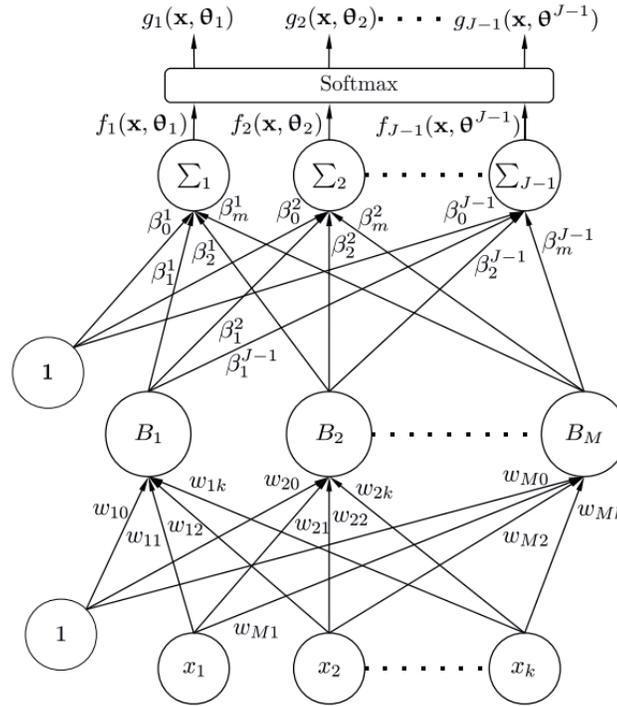


Figura 3.16 Uso de función *Softmax*

## 3.2 Optimización por Enjambre de Partículas

El algoritmo de Optimización por enjambre de Partículas o PSO debido a sus siglas en inglés (*Particle Swarm Optimization*) fue desarrollado por Kennedy y Eberhart en 1995 [32], basando su inspiración en el comportamiento de sistemas biológicos sociales como lo son las bandadas de pájaros, cardúmenes de peces e insectos entre otros. Ya que trata de simular el comportamiento de estos sistemas biológicos sociales en situaciones tales como búsqueda de alimento o de lugar de cobijo. Para ello se considera que cada individuo tiene un comportamiento determinado tanto por su búsqueda individual como por la búsqueda grupal. De este modo se logra llegar a una analogía en la cual se tiene un enjambre de partículas que simulan a los individuos en un sistema biológico social, estas partículas se desplazan en un espacio en el cual buscan soluciones. La calidad de las soluciones se determina mediante la función *fitness*.

Así, consideramos lo siguiente:

- Un espacio de N dimensiones para cada partícula i, en el cual se buscarán soluciones.
- Cada partícula conoce su propio  $X_i$ ,  $V_i$  y  $P_i$ .
- La posición actual  $X_i = [x_{i1}, \dots, x_{ij}, \dots, x_{iN}]$ , donde  $x_{ij}$  representa la posición actual de la i-ésima partícula en la j-ésima dimensión.
- La velocidad  $V_i = [v_{i1}, \dots, v_{ij}, \dots, v_{iN}]$  con la que se llegó a dicha posición, donde  $v_{ij}$  representa la velocidad en la j-ésima dimensión con la cual la i-ésima partícula llegó a la posición actual.
- La mejor posición que ha tenido  $P_i = [p_{i1}, \dots, p_{ij}, \dots, p_{iN}]$ , conocida como *mejor personal*, donde  $p_{ij}$  representa la posición de la i-ésima partícula en la j-ésima dimensión.
- La mejor posición que se ha encontrado en el enjambre  $G = [g_1, \dots, g_j, \dots, g_N]$ , conocida como *mejor global*, donde  $g_j$  representa la mejor posición encontrada en la dimensión j. Esta posición es conocida por todas las partículas.
- El coeficiente de inercia  $\omega$ , el cual regula el impacto de la velocidad anterior en la nueva velocidad y cuyo valor es aleatorio y se encuentra normalmente en el intervalo  $[0.5, 1]$ . Un valor alto ayudará a una exploración global, mientras que un valor bajo favorecerá a una exploración local
- Parámetro cognitivo  $C_1$ , el cual indica la influencia máxima de la mejor experiencia de la partícula en su nueva velocidad.
- Parámetro social  $C_2$  que indica la influencia máxima de la información social en la nueva velocidad de la partícula.
- Tanto a  $C_1$  como  $C_2$  se les suele asignar un valor igual a 2.
- $\text{rand}()$  representa un valor aleatorio en el intervalo  $[0, 1]$ .
- Si  $V_{\max}$  es muy pequeño, las partículas explorarán el espacio de solución muy lentamente pudiendo quedar atrapados en un óptimo local.
- La velocidad de cada partícula puede estar dentro del intervalo  $[-V_{\max}, V_{\max}]$ . Si la velocidad obtenida está fuera de este rango se ha de aplicar la siguiente función:

$$v_{ij} = \text{sign}(v_{ij}) * V_{\max} \quad (3.2.1)$$

En la cual:

$$\text{sign}(x) = \begin{cases} +1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \quad (3.2.2)$$

- El número máximo de iteraciones puede estar determinado por un número establecido de iteraciones, según se alcance cierto número fijo de iteraciones sin presentar mejoras en la solución o al alcanzar un cierto valor *fitness*.
- Con valores altos,  $0.75 < \omega < 1$  y  $2 < C_1, C_2 < 4$ . Se favorece a buscar soluciones en un espacio de búsqueda amplio.
- Con valores bajos,  $0.4 < \omega < 0.75$  y  $0.1 < C_1, C_2 < 2$ . Se favorece a buscar soluciones en solo un rango del espacio total de búsqueda.

Ya teniendo claro lo anterior, se observa la forma en que funciona esta meta heurística mediante los siguientes pasos [1]:

1. Inicializar el conjunto de partículas con posiciones ( $X$ ) y velocidades ( $V$ ) aleatorias.
2. Evaluar la función *fitness* para cada partícula.
3. Comparar la función *fitness* actual de cada partícula con su mejor personal ( $P$ ). Si  $X_i$  es mayor que  $P_i$ , entonces  $P_i$  toma el valor  $X_i$ . Si  $X_i$  es mayor que  $G_i$ , entonces  $G_i$  toma el valor  $X_i$ .
4. Ajustar la posición actual y la velocidad de cada partícula [33] según las siguientes ecuaciones:

$$v_{ij}^{t+1} = \omega \times v_{ij}^t + C_1 \times \text{rand}(\ ) \times (p_{ij}^t - x_{ij}^t) + C_2 \times \text{rand}(\ ) \times (g_j^t - x_{ij}^t) \quad (3.2.3)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (3.2.4)$$

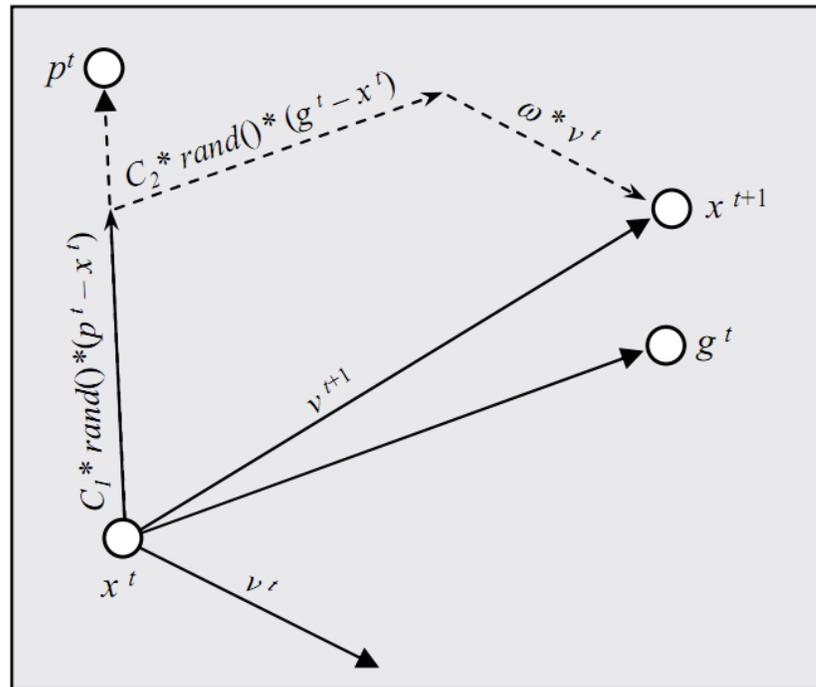


Figura 3.17 Actualización de una partícula [33]

5. Si se cumple con el número de iteraciones establecido o si se alcanza el *fitness* deseado se detiene el proceso, de lo contrario se va al paso 2.

### 3.2.1 VARIANTES DE PSO

No es de extrañarse que desde su invención en 1995 el algoritmo PSO haya sido objetivo de una serie de cambios desde su primera propuesta, estos cambios buscan aliviar el

estancamiento en la optimización. A continuación se mostrara algunas de las variantes de PSO.

### 3.2.1.1 PSO Discreto

PSO discreto [34] [35] [36]: si bien la mayoría de los algoritmos de optimización de enjambre de partículas están pensados para buscar soluciones en un dominio continuo, hay variantes que están pensadas para espacios discretos. La primera variante para un dominio discreto fue PSO binario [37]. En esta se considera la posición discreta, mientras su velocidad es continua. El componente j-ésimo del vector de velocidad de la partícula es usado para calcular la probabilidad con la cual el j-ésimo componente del vector de posición de la partícula toma el valor uno. La velocidad se actualiza según la ecuación 3.2.3, mas la posición es actualizada usando la siguiente ecuación:

$$x_{ij}^{t+1} = \begin{cases} 1 & \text{si } r < \text{sig}(v_{ij}^{t+1}) \\ 0 & \text{si } r \geq \text{sig}(v_{ij}^{t+1}) \end{cases} \quad (3.2.5)$$

Donde  $x_{ij}$  es la j-ésima componente del vector de posición de la partícula i, r es un número aleatorio en el intervalo [0,1] y  $\text{sig}(v_{ij})$  que representa la probabilidad de que el bit j-ésimo de la partícula i sea uno,  $\text{sig}(v_{ij})$  está dada por la siguiente ecuación:

$$\text{sig}(v_{ij}) = \frac{1}{1 + e^{-v_{ij}}} \quad (3.2.6)$$

### 3.2.1.2 PSO Coeficiente de Constricción

*Constricted coefficients PSO* [35] [36] [38]: Este método fue planteado por Clerc [39] en el año 2000 para controlar el cambio en la magnitud de la velocidad. Para ello se utiliza la ecuación 3.2.4 y la actualización de velocidad se calcula según la siguiente ecuación:

$$v_i^{t+1} = \lambda \times (v_i^t + C_1 \times \text{rand}(\ ) \times (p_i^t - x_i^t) + C_2 \times \text{rand}(\ ) \times (g^t - x_i^t)) \quad (3.2.7)$$

Donde  $\lambda$  es el coeficiente de constricción, cuyo valor está determinado por la siguiente ecuación:

$$\lambda = \frac{2k}{[2 - C - \sqrt{(C^2 - 4 \times C)}]} \quad (3.2.8)$$

Donde  $C=C_1+C_2$  y  $C>4$ . Se eligen estos valores con la intención de asegurar que el enjambre converja a una pequeña región del espacio de búsqueda, comúnmente k es uno y el valor de  $\lambda$  es 0.7298, esto al tomar  $C_1 = C_2 = 2.05$ .

Este algoritmo está pensando para que las partículas converjan en el tiempo, esto al reducirse la amplitud conforme se focaliza en el mejor personal y en el mejor global. La partícula oscilara alrededor del mejor personal y el mejor global, si estos dos están cerca las

partículas desarrollaran una búsqueda local, si están lejos, las partículas desarrollaran una búsqueda global.

### 3.2.1.3 PSO *Bare bones*

PSO Bare Bones [35] [36] [40]: es una versión de PSO planteada por Kennedy el 2003, en ella las reglas de actualización de la posición y la velocidad se reemplazan por un procedimiento que muestree una función de densidad de probabilidad paramétrica.

Para ello se usan las siguientes ecuaciones:

$$x_{ij}^{t+1} = N(\mu_{ij}^t, \sigma_{ij}^t) \quad (3.2.9)$$

$$\mu_{ij}^t = \frac{p_{ij}^t + g_j^t}{2} \quad (3.2.10)$$

$$\sigma_{ij}^t = |p_{ij}^t - g_j^t| \quad (3.2.11)$$

Donde  $N(\mu, \sigma)$  representa un número según la distribución normal en la cual  $\mu$  es la media y  $\sigma$  es la desviación estándar,  $p_{ij}$  es el mejor personal de la partícula  $i$ -ésima en la dimensión  $j$ -ésima y  $g$  es el mejor global en la dimensión  $j$ -ésima.

### 3.2.1.4 Linearly Decreasing Weight Particle Swarm Optimization

*Linearly Decreasing Weight Particle Swarm Optimization* o *LDWPSO* [41] [42] [43]: fue propuesto por Shi y Eberhart en 1998 [44], en él se consideran las ecuaciones 3.3.4 y 3.2.3 de manera que el factor de inercia va disminuyendo su valor conforme aumenta el número de iteraciones. Esto mejora el desempeño con respecto al PSO original al balancear las habilidades de búsquedas globales y locales. Para lograr esto el peso de inercia del LDWPSO ( $\omega_{LDW}$ ) decrece linealmente de 0.9 a 0.4, su valor en cada iteración estará dado por la siguiente ecuación:

$$\omega_{LDW} = (\omega_{\max} - \omega_{\min}) \times \frac{\text{iteración}_{\max} - \text{iteración}_i}{\text{iteración}_{\max}} + \omega_{\min} \quad (3.2.12)$$

Donde  $\omega_{\max}$  es 0.9,  $\omega_{\min}$  es 0.4 y  $\text{iteración}_{\max}$  es el máximo de iteraciones permitidas.

### 3.2.1.5 Center PSO

*Center PSO* [42] [43]: es una modificación del algoritmo LDWPSO, surgiendo como resultado de implantar una partícula central en el enjambre, esta partícula no tiene una velocidad explícita y se ubica en el centro del enjambre en cada iteración, esto porque el centro del enjambre generalmente presenta un buen *fitness*.

Para este algoritmo una vez que las N-1 partículas actualicen su posición como usualmente se hace en el PSO original, una partícula actualiza su posición según la fórmula:

$$x_{cj}^{t+q} = \frac{1}{N-1} \sum_{i=1}^{N-1} x_{ij}^{t+1} \quad (3.2.13)$$

A diferencia de las otras partículas, esta no tiene velocidad, mas es considerada para todas las operaciones como una partícula más del enjambre.

### 3.2.1.6 Simple PSO

*Simple PSO* [42]: Se diferencia del PSO en que la velocidad de la partícula expresada en la ecuación 3.2.3 está dada por la siguiente ecuación:

$$v_{ij}^{t+1} = C_1 \times rand() \times (p_{ij}^t - x_{ij}^t) + C_2 \times rand() \times (g_j^t - x_{ij}^t) \quad (3.2.14)$$

Por lo que se puede decir que la partícula no tiene memoria sobre la velocidad anterior.

### 3.2.1.7 PSO Quantum

Este algoritmo también llamado QPSO [45] fue planteado basándose en la lógica de la mecánica cuántica.

En términos de mecánica clásica la trayectoria de la partícula la podemos determinar según el vector de posición  $X_i$  y el vector de velocidad  $V_i$ , esto en la mecánica Newtoniana, pero no en el caso de la mecánica cuántica. Según la mecánica cuántica no tiene sentido tratar de determinar la trayectoria, ya que  $X_i$  y  $V_i$  no pueden ser determinadas simultáneamente según el principio de incertidumbre. Por lo tanto si se considera el caso en que la partícula de PSO posea un comportamiento cuántico, el algoritmo PSO debe ser trabajar de otra forma.

El estado de una partícula es representado por la función de onda  $\psi(x,t)$  (*Schrodinger equation*), en vez de la posición y la velocidad. Teniendo en cuenta que la posición y la velocidad no pueden ser determinadas simultáneamente, entonces la probabilidad de que la partícula aparezca en la posición  $X_i$  dependerá de la función de densidad de probabilidad  $|\psi(x,t)|^2$  y la forma en la cual depende de los campos potenciales se encuentra en [46].

Al utilizar el método de Monte Carlo, la ecuación de movimiento está determinada por:

$$\begin{cases} x_i(t+1) = p + \beta \times |Mbest_i - x_i(t)| \times \ln\left(\frac{1}{u}\right) & \text{si } k \geq 0.5 \\ x_i(t+1) = p - \beta \times |Mbest_i - x_i(t)| \times \ln\left(\frac{1}{u}\right) & \text{si } k < 0.5 \end{cases} \quad (3.2.15)$$

Donde  $\beta$  es un parámetro llamado expansión-contracción el cual toma un valor en el intervalo  $[0,1.7]$ ,  $u$  y  $k$  son valores generados acordes a la distribución de probabilidad uniforme entre el rango  $[0,1]$ .

$Mbest_i$  es un vector que tiene las medidas de las mejores posiciones encontradas por las partículas, su valor está dado por:

$$Mbest = \frac{1}{N} \sum_{i=1}^N p_i = \frac{1}{N} \sum_{i=1}^N p_{i1}, \frac{1}{N} \sum_{i=1}^N p_{i2}, \dots, \frac{1}{N} \sum_{i=1}^N p_{id} \quad (3.2.16)$$

Donde  $N$  es el número de partículas del enjambre y  $p_i$  corresponde al mejor personal, es decir se promedia el mejor personal de todas las partículas en la dimensión correspondiente.

Ahora para determinar  $p$  usamos:

$$p = \varphi \times p_{id} + (1 - \varphi) \times p_{gd} \quad (3.2.17)$$

Donde  $\varphi$  es un valor aleatorio en el intervalo  $[0,1]$ .

### 3.2.1.8 Métodos de Mejoramiento de PSO para Optimización Global

Métodos de mejoramiento de PSO para optimización global [42]: hasta ahora se han visto variantes de PSO, todas ellas se basan en PSO, pero a continuación se mostrará un método para mejorar la eficiencia y velocidad, el cual puede ser aplicado en la mayoría de las variantes de PSO. Este método considera las tres siguientes modificaciones:

- 1) Una nueva regla de detención.
- 2) Una verificación similar.
- 3) Una aplicación condicional de algunos métodos de búsqueda local.

Una nueva regla de detención: esta nueva regla se crea con el fin de disminuir el número de iteraciones, esto se logra calculando en cada iteración la variación del mejor valor descubierto ( $v$ ), deteniéndose cuando  $v$  es menor o igual a  $s$ , donde  $s$  el mejor valor descubierto cuando un nuevo mejor valor fue recientemente descubierto. La razón de esta regla es que de no haber progreso en un número de iteraciones, podemos creer que hemos encontrado el mínimo global, por lo que debemos detener las iteraciones.

Una verificación similar: esta modificación evita que la función objetivo evalúe el punto  $x_i$  aun cuando este no cambie en la última iteración. Esto controlando los cambios en el punto  $x_i$ , por ejemplo, evaluar  $f(x_i)$  si y solo si  $|x_i^{t+1} - x_i^t| \geq \epsilon$ , donde  $\epsilon$  representa un pequeño número positivo cercano a cero.

Aplicación de búsqueda local: un procedimiento de búsqueda local usado es una variante de BFGS.

### 3.3 Método de Entrenamiento

Las redes neuronales artificiales *feed-forward* poseen diversos métodos de entrenamiento, uno de los más conocidos es el de Retro Propagación, el cual es un método supervisado de entrenamiento, por lo que debe conocerse la salida deseada de la red para cada patrón de entrada, otro método de entrenamiento es mediante algoritmos evolutivos como PSO, el cual es utilizado en esta investigación. Por lo general una forma de entrenar redes neuronales es mediante el error cuadrático medio, pero este es utilizado mayormente para fines predictivos, su fórmula está dada por:

$$E = \frac{1}{N} \times \sum_{i=1}^N (d_i(n) - y_i(n))^2 \quad (3.3.1)$$

Donde N es el número de patrones para entrenar la red,  $d_i$  es el valor deseado para el  $i$ -ésimo patrón de entrenamiento e  $y_i$  es el valor que entrega la red para el  $i$ -ésimo patrón de entrenamiento.

Otra forma de entrenar es utilizando la entropía cruzada, la cual está dada por la fórmula 3.3.2, de manera de buscar E cercanos a 0. La siguiente ecuación considera que la red tiene una sola salida, en donde  $d$  es la salida deseada de la red e  $y$  es la salida de la red.

$$E = -\frac{1}{N} \times \sum_{i=1}^N (d_i(n) \times \ln(y_i(n)) + (1 - d_i(n)) \times \ln(1 - y_i(n))) \quad (3.3.2)$$

Para calcular la entropía cruzada de una red de múltiples salidas se ocupa la ecuación (3.3.3), donde  $d$  es la salida deseada de la red e  $y$  es la salida de la red, además  $y_l$  representa las clases de salidas de la red desde 1 hasta J y  $\theta$  es el vector  $(\theta_1, \dots, \theta_j)$ .

$$E = -\frac{1}{N} \times \sum_{i=1}^N \left[ -\sum_{l=1}^J d_l(n) \times y_l(x(n), \theta(l)) + \log \sum_{l=1}^j e^{y_l(x(n), \theta(l))} \right] \quad (3.3.3)$$

Entonces se requiere de métricas acordes a un modelo de clasificación para determinar el rendimiento del modelo, para lo cual es preciso conocer primero los siguientes términos:

- Verdaderos Positivos o VP: también conocidos como éxitos, corresponde al número de personas clasificadas por el modelo como lesionadas y resultan realmente lesionadas.
- Falso Positivo o FP: también conocidos como falsas alarmas, corresponde al número de personas clasificadas por el modelo como lesionadas, pero que resultan realmente ilesas.
- Verdaderos Negativos o VN: también conocidos como rechazos correctos, corresponde a las personas clasificadas por el modelo como ilesas y resultan realmente ilesas.

- Falsos Negativos o FN: corresponde al número de personas clasificadas por el modelo como ilesas, pero que resultan realmente lesionadas.

Esto se puede visualizar en la siguiente tabla, en donde se señalan los tipos de error.

**Tabla 3.3 Matriz de Confusión**

		Resultado Real	
		Lesionadas	Ilesas
Resultado obtenido por el modelo	Lesionadas	VP	FP, error tipo II
	Ilesas	FN, error tipo I	VN

Teniendo claro lo anterior es que se pueden establecer las siguientes métricas para determinar el rendimiento del modelo de clasificación:

- Sensibilidad: indica la probabilidad de clasificar a un individuo como lesionado, es decir el porcentaje de personas que son clasificadas efectivamente como lesionadas. La sensibilidad está definida por la siguiente ecuación:

$$Sensibilidad = \left( \frac{VP}{VP + FN} \right) \quad (3.3.4)$$

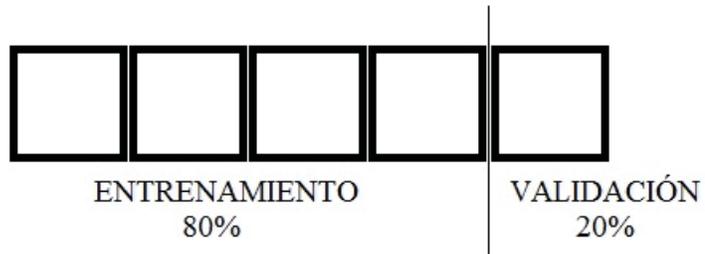
- Especificidad: indica la probabilidad de que un individuo ileso sea realmente clasificado como ileso. La especificidad está definida por la siguiente ecuación:

$$Especificidad = \left( \frac{VN}{VN + FP} \right) \quad (3.3.5)$$

- Exactitud: indica el porcentaje de personas bien clasificadas, tanto lesionadas como ilesas. La exactitud está definida por la siguiente ecuación:

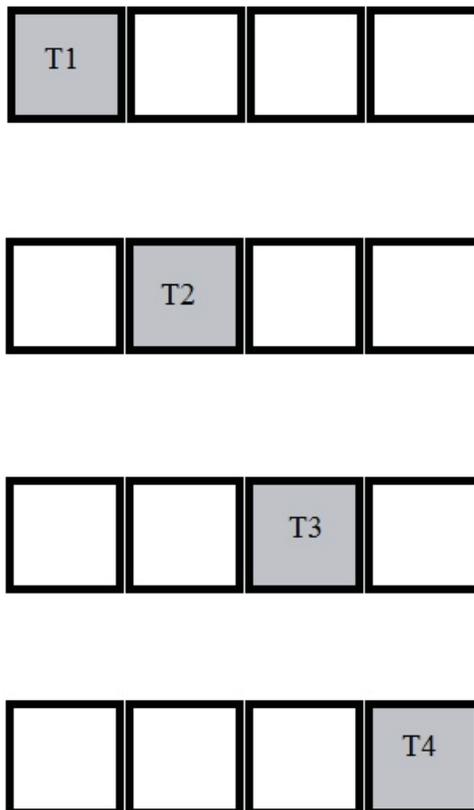
$$Exactitud = \left( \frac{VP + VN}{VP + VN + FP + FN} \right) \quad (3.3.6)$$

Las métricas explicadas anteriormente pueden ser utilizadas en conjunto con la técnica de Validación Cruzada. Este método consiste en dividir los datos en dos partes, una para pruebas y otra para entrenamiento de la red neuronal, un ejemplo es usar el 80% de los datos para entrenamiento y un 20% para validación, como se aprecia en la figura 3.18.



**Figura 3.18 División de Datos Para Validación Cruzada**

Los datos de entrenamiento se dividen en  $k$  partes, para este ejemplo en cuatro, donde se va dejando un conjunto de datos para hacer entrenamiento y otro para pruebas como se muestra en la figura 3.18.



**Figura 3.19 División de Datos de entrenamiento Para Validación Cruzada**

Luego se calcula el promedio de los *fitness* de los  $T$ , en este caso de  $T_1$ ,  $T_2$ ,  $T_3$  y  $T_4$  como se puede apreciar en la figura 3.19, este *fitness* será utilizado como medida del entrenamiento. Además se guardarán los valores de los pesos que den mejores resultados en el entrenamiento.

## 4. Descripción de Datos en Estudio

Para el estudio se utilizarán los datos suministrados por la Escuela de Ingeniería de Transporte de la PUCV, cuya escuela tiene nexos con CONASET, estos datos se encuentran almacenados en plantillas Excel.

Los datos se obtienen mediante su registro en plantillas SIEC2 (Anexo A) para cada accidente, siguiendo un instructivo para posteriormente ser ingresados en una base de datos de CONASET.

Los datos son separables en tres entidades, cada una con sus respectivos atributos para fines de registro:

1. Accidentes: identificador del accidente, fecha, hora de accidente, comuna, causas, tipo de accidente, estado atmosférico, condición calzada, tipo de calzada, estado calzada, urbano/rural, y la información que indica la cantidad de ilesos, leves, graves, menos graves y muertos.
2. Persona: identificador del accidente, sexo, calidad, edad, y resultado.
3. Vehículo: identificador del accidente, servicio, y el tipo de vehículo.

Como se indica en la siguiente figura, en cada accidente puede haber una o muchas personas involucradas y/o uno o muchos vehículos involucrados.

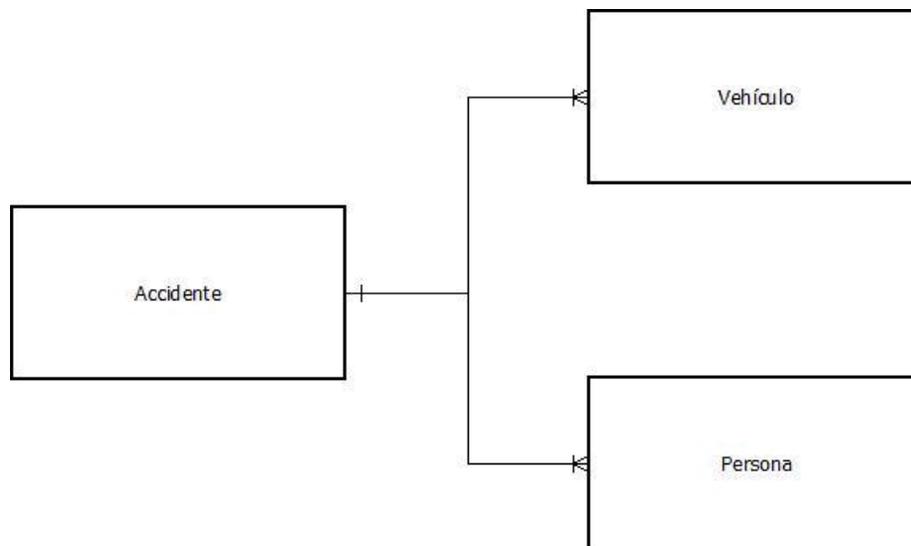


Figura 4.1 Entidades base de datos CONASET

A continuación se detallan los atributos propios de cada entidad [4].

1. Accidentes. Entidad correspondiente a los accidentes ocurridos en una determinada fecha y lugar geográfico. Posee una serie de atributos, estos son:

- Idaccidente: identificador único y correlativo por cada accidente ocurrido en todo Chile.
- Fecha: fecha del suceso.
- Hora: hora del suceso.
- Región: región del suceso.
- Comuna: comuna de la región en donde ocurrió el accidente.
- Tipo de accidente: existen diez tipos accidentes, estos son:
  - Atropello.
  - Caída.
  - Colisión: dentro de estas se presentan las colisiones frontal, lateral, por alcance y perpendicular.
  - Impacto con animal.
  - Choque con objeto: dentro de estos se presentan los choques con objeto frontal, lateral y posterior.
  - Choque con vehículo detenido: dentro de estos se presentan los choque con vehículo detenido frente/frente, frente/lado, frente/posterior, lado/frente, lado/lado, lado/posterior, posterior/frente, posterior/lado y posterior/posterior.
  - Volcadura.
  - Incendio.
  - Descarrilamiento.
  - Otro tipo.
- Causa: corresponde al origen del accidente, existiendo dieciocho tipos, estas son:
  - Fallas mecánicas: dentro de estas se encuentran las fallas del tipo frenos, dirección, eléctrica, suspensión, neumáticos, motor y carrocería.
  - Adelantamiento: dentro de estas se encuentran los adelantamientos sin el espacio o tiempo suficiente, sin efectuar la señal respectiva, por la berma, sobrepasando línea continua, en cruce, curva, cuesta, puente, conducción, bajo la influencia del alcohol, bajo la influencia de drogas o estupefacientes, contra el sentido del tránsito, en estado de ebriedad, físicas deficientes (cansancio, sueño), por izquierda eje calzada, no atento condiciones del tránsito momento, sin mantener distancia razonable ni prudente y cambiar sorpresivamente pista de circulación.
  - No respetar derecho a paso: dentro de estos se encuentran los derechos al peatón y vehículo.
  - Pasajero: dentro de estos se presentan las situaciones como si sube o desciende de vehículo en movimiento, viaja en pisadera de vehículo, imprudencia y ebriedad.
  - Peatón: dentro de estos se presentan las situaciones: permanece sobre la calzada, cruza calzada forma sorpresiva o descuidada, imprudencia, ebriedad, cruza calzada fuera paso peatones y cruza camino o carretera sin precaución.

- Señalización: dentro de ellas están: mal instalada o mantenida forma defectuosa, desobedecer luz roja de semáforo, desobedecer indicación carabinero servicio, desobedecer señal ceda el paso, desobedecer señal pare, desobedecer otra, semáforo mal estado o deficiente y desobedecer luz intermitente semáforo.
  - Velocidad: dentro de ellas se encuentran: mayor que máxima permitida, no razonable ni prudente, no reducir cruce de calles, cumbre, curva, etc., exceso en zona restringida, menor que mínima establecida.
  - Carga: dentro de ellas se encuentran: mayor que la autorizada vehículo, obstruye visual conductor, escurre a la calzada, sobresale estructura vehículo.
  - Virajes indebidos.
  - Animales sueltos en vía pública.
  - Vehículos en retroceso, conducir.
  - Vehículos en pana sin señalización o deficiente.
  - Pérdida control vehículo.
  - Suicidio.
  - Causas no determinadas.
  - Otras causas.
  - Fuga por hecho delictual.
- Ubicación relativa: Existen diecisiete ubicaciones, estas son:
    - Cruce con semáforo funcionando, cruce sin señalización, enlace a desnivel, plaza de peaje, acera o berma, enlace a nivel, tramo de vía curva vertical, túnel, cruce con señal "pare", cruce regulado por carabinero, tramo de vía recta, acceso no habilitado, tramo de vía curva horizontal, rotonda, otros no considerados, cruce con señal "ceda el paso", cruce con semáforo apagado y puente.
- Estado Atmosférico: Existen seis estados atmosféricos, estos son:
    - Despejado, nublado, lluvia, llovizna, neblina, nieve.
- Tipo calzada: Existen seis tipos de calzada, estos son:
    - Concreto, asfalto, adoquín, mixto, ripio y tierra.
- Estado Calzada: estas son:
    - Bueno, regular, malo.
- Condición Calzada: estos son nueve:
    - Seco, húmedo, mojado, con barro, con nieve, con aceite, escarcha, gravilla y otros.

- Muertos: cantidad de muertos involucrados en el accidente.
  - Graves: cantidad de graves involucrados en el accidente.
  - Menos Graves: cantidad de menos graves involucrados en el accidente.
  - Leves: cantidad de leves involucrados en el accidente.
  - Ilesos: cantidad de ilesos involucrados en el accidente.
2. Personas: Entidad asociada a las personas involucradas en un accidente en particular. Un accidente por lo menos posee una persona involucrada, ya sea conductor, pasajero o peatón. Los atributos relacionados a esta entidad son:
- Calidad: se subdividen en:
    - Peatón, conductor, pasajero.
  - Sexo.
  - Resultado: corresponden a cinco estados en los que pueden quedar las personas:
    - Muerto, grave, menos grave, leve, ileso.
3. Vehículos: Entidad asociada a los vehículos involucrados en un accidente en particular. Un accidente por lo menos presenta uno o más vehículos. Los atributos y características asociadas a esta entidad son:
- Tipo vehículo: dentro de los tipos se encuentran treinta tipos de vehículos, estos:
  - Bus/ taxi bus, minibús, trolebús, automóvil, camioneta, jeep, furgón, ambulancia, camión simple, camión simple con remolque, tracto-camión, tracto-camión con remolque, carro bomba, carro transporte de valores, remolque/semi remolque, motocicleta, motoneta/bicimoto, moto arenera, bicicleta, tracción animal, carro tracción humana, tractor, maquinaria agrícola, maquinaria movimiento tierras, maquinaria industrial, patín/patineta, patín motorizado, ferrocarril, dado a la fuga, otros no clasificados.
  - Servicio: dentro de los tipos de servicios encontramos 17 tipos de servicio, estos son:
    - Carabineros, fiscal, particular, transporte escolar, taxi básico, taxi colectivo urbano, taxi colectivo rural, bomberos, salud, locomoción colectivo urbano, locomoción colectivo rural, servicio interurbano, servicio internacional, carga normal, carga peligrosa, dado a la fuga, otros sin especificar.

## 5. Desarrollo de los Modelos

Para llegar a la obtención de los modelos se ha de seguir una serie de pasos, estos se detallan a continuación:

1. Tratamiento de datos: en la cual se hace un pre-proceso de los datos en el cual se elimina redundancia e inconsistencias para luego normalizarles facilitando la labor posterior.
2. Elección de red neuronal artificial y algoritmo de entrenamiento: en esta etapa se seleccionan los tipos de redes usados para realizar los diversos modelos, para esta investigación se utilizarán MLP y una adaptación de MLP la cual utilizará softmax en su capa de salida, en cada una de ellas se probarán diversas funciones como se verá más adelante.
  - a. Selección de algoritmo de entrenamiento: el algoritmo de entrenamiento a utilizar será LDWPSO, QPSO y/o LDWQPSO.
  - b. Elección de topologías: los nodos de entrada de la red son 12, los ya mencionados anteriormente. Los nodos de la capa oculta se elegirán según LDWPSO, QPSO y/o LDWQPSO. La capa de salida presenta uno o dos nodos de salida.
3. Entrenamiento de la red: la red se entrena mediante LDWPSO, QPSO y/o LDWQPSO de manera de obtener los pesos de la red, para ello se utilizará un determinado número de datos como mostrará más adelante, por ejemplo un 80% de los datos.
4. Testeo de la red: se obtienen los resultados una vez ya entrenada la red, para ello se utilizará un determinado número de datos, los cuales no haya sido utilizados para el entrenamiento, por ejemplo un 20% de los datos.
5. Comparación de resultados: en la etapa final se compararán los resultados obtenidos por los diversos modelos mediante métricas especificadas más adelante.



Figura 5.1 Pasos para desarrollo del modelo

## 5.1 Tratamiento de Datos

Los datos empleados para la investigación fueron proporcionados por Felix Castro. Estos datos en un principio fueron obtenidos de la CONASET por la Escuela de Ingeniería de transporte de la PUCV en formato Excel. Posteriormente se traspasaron a un motor de base de datos MySQL, en donde se crearon las tres entidades mencionadas en la figura 5.1 (Accidente, Persona y Vehículo), descartándose los registros que poseían campos nulos o pertenecientes a regiones diferentes a Valparaíso, ya que en esta se basa la investigación para la clasificación de accidentes de tránsito, una vez hecho esto se importaron los datos a una matriz alojada en un archivo de Matlab data.mat. De esta manera se trabajó con los datos recolectados desde el año 2003 al 2009.

### 5.1.1 Selección de Datos de Entrada y Codificación.

Los atributos de entrada para los modelos de clasificación propuestos, los cuales clasifican el estado (lesionadas o ilesas) de las personas involucradas en accidentes de tránsito, fueron escogidos de los atributos utilizados en el trabajo [4] realizado por Félix Castro de la Escuela de Ingeniería Informática de la PUCV fueron:

**Tabla 5.1 Atributos de entrada del modelo**

Número	Atributo	Ejemplo
1	Comuna	..., 5102,5603,...
2	Urbano/Rural	1 o 2
3	Estado Atmosférico	1,2 o 3
4	Hora	0,1,...,24
5	Causa	1,...,99
6	Tipo Calzada	1,2,3,4,5 o 6
7	Estado Calzada	1,2 o 3
8	Condición Calzada	0,1,2,3,4,5,6,7,8 o 9
9	Tipo de Accidente	10,...,99
10	Calidad	1,2 o 3
11	Sexo	1 o 2
12	Edad	0,1,3,...
13	Resultado	1,2,3,4 o 5

Como se aprecia todos los atributos que no eran atributos numéricos fueron reemplazados por atributos numéricos, esto para poder ser tratados por las técnicas propuestas, es así que se realizaron cambios del siguiente tipo:

- Estado Calzada:
  - Bueno 1
  - Malo 2
  - Regular 3

- Comuna:
  - Valparaíso 5101
  - Quilpué 5106

Respecto al atributo “Resultado”, este fue llevado a valor numérico, para lo cual se consideró el siguiente cambio:

- Resultado
  - Muerto 1
  - Grave 2
  - Menos Grave 3
  - Leve 4
  - Ileso 5

Es el campo “Resultado” el tomado en cuenta para obtener las etiquetas que permitirán al modelo clasificar en:

**Tabla 5.2 Formación de clases [4]**

Clase	Atributos	Etiqueta
Persona Lesionada	Grave, Menos Grave y Leve	1
Persona Ilesa	Ileso	-1

Como se aprecia en la tabla anterior, los casos de personas en estado “Muerto” no son considerados por el modelo. Además el atributo resultado no es usado como una entrada al modelo, ya que este es remplazado por una etiqueta utilizada para entrenar y corroborar los resultados del modelo.

### 5.1.2 Normalización de los datos

Los datos normalizados permiten por lo general obtener mejores resultados, esto al acotar el rango en el cual se encuentran los datos, es por ello que los datos mencionados anteriormente fueron normalizados mediante la siguiente fórmula aplicada a cada atributo por separado:

$$z = factor * \left[ \frac{2 \times x - (\max(x) + \min(x))}{\max(x) - \min(x)} \right] \quad (5.1.1)$$

Donde el factor tiene un valor de 0.8 o 0.9,  $\max(x)$  representa el máximo valor existente en un atributo,  $\min(x)$  representa el mínimo valor existente en un atributo.

### 5.1.3 Comparación de modelos

Para comparar los modelos propuestos se consideran tres variables: exactitud, especificidad y sensibilidad, estas son contrastadas entre modelos mediante la ganancia, de esta forma es posible determinar si el modelo permitió una ganancia positiva o negativa (perdida). La fórmula de ganancia está dada por la siguiente ecuación:

$$G(A, B) = \frac{A - B}{A} \times 100\% \quad (5.1.2)$$

Donde G representa la ganancia de la A sobre la B de manera porcentual.

#### **5.1.4 Entorno de trabajo**

El entorno de trabajo considera las características de hardware y software utilizadas para desarrollar e implementar los modelos de clasificación.

El hardware utilizado consta de:

- Procesador Intel Core i3 2,27 GHz.
- Memoria RAM 4,0 GB.

El software utilizado consta de:

- Windows 7 Home Premium de 64 bits.
- Matlab 7.6.0.324 R2008a.

## 6. Primeros Modelos propuestos

### 6.1 MLP

Para determinar los modelos se siguieron los siguientes pasos una vez ya procesados los datos:

- Elegir las funciones de activación a utilizar.
- Determinar el número de nodos ocultos.
- Entrenar red neuronal mediante algoritmo LDWPSO y evaluar que red presenta mejores resultados.

#### 6.1.1 Elegir las funciones de activación a utilizar

Se realizó la elección de dos funciones de activación para la capa oculta y de salida a fin de obtener cuatro redes neuronales, la cuales se diferencian en el número de nodos ocultos y sus respectivas funciones de activación, es así que se obtienen las siguientes cuatro redes neuronales:

- Red neuronal con función de activación tangente hiperbólica para la capa oculta y de salida. Nos referiremos a ella como **red neuronal t-t** a fin de facilitar su reconocimiento.
- Red neuronal con función de activación sigmoideal para la capa oculta y tangente hiperbólica para la capa de salida. Nos referiremos a ella como **red neuronal s-t** a fin de facilitar su reconocimiento.
- Red neuronal con función de activación sigmoideal para capa oculta y de salida. Nos referiremos a ella como **red neuronal s-s** a fin de facilitar su reconocimiento.
- Red neuronal con función de activación tangente hiperbólica para la capa oculta y sigmoideal para la capa de salida. Nos referiremos a ella como **red neuronal t-s** a fin de facilitar su reconocimiento.

#### 6.1.2 Determinar la Cantidad de Nodos Ocultos

Ya habiendo elegido las funciones de activación para cada una de las redes tenemos 4 tipos de redes, cada una de ellas compuesta por 12 nodos de entrada y 1 nodo de salida por lo que queda determinar el número de nodos ocultos y entrenar los pesos sinápticos de las redes. Teniendo esto en mente se determinará entonces el número de nodos ocultos.

Para determinar el número de nodos ocultos se utilizaron 3000 datos de los cuales se utilizó el 80% para obtener el desempeño durante el entrenamiento con 300 iteraciones para así realizar 10 pruebas con los datos de prueba, cada una de ellas desde 1 nodo oculto hasta 25 nodos ocultos. Las redes fueron entrenadas con LDWPSO, posteriormente se obtuvo la media aritmética para así elegir el número de nodos ocultos que presentó el menor error determinado por la siguiente ecuación:

$$\text{Error} = 1 - \text{Exactitud} = 1 - \left( \frac{VP+VN}{VP+VN+FP+FN} \right) \quad (6.1.1)$$

Los resultados para la obtención del número de nodos ocultos para cada red propuesta se pueden visualizar en la tabla 6.1, la cual muestra el número de nodos ocultos que presentó el menor error promediado de diez pruebas realizadas desde 1 nodo oculto hasta 25.

**Tabla 6.1 Mejores resultados en elección de nodos ocultos, primeros modelos**

Red Neuronal	Nº de Nodos Ocultos	Max.	Min.	Desv.	Prom.
t-t	4	0,2121	0,2008	0,0039	0,2053
s-t	4	0,2025	0,1842	0,0055	0,1945
s-s	5	0,2083	0,1925	0,0054	0,2013
t-s	4	0,2171	0,1975	0,0061	0,2041

Los resultados de la tabla 6.1 son los mejores resultados de las tablas B.1, B.2, B.3 y B.4 perteneciente al anexo B, en ella se aprecia como varía el error (ecuación 6.1.1) promediado de diez pruebas considerando desde un nodo oculto hasta veinti cinco para cada tipo de red.

Con los datos obtenidos de las tablas B.1, B.2, B.3 y B.4 perteneciente al anexo B, se puede determinar que las redes neuronal t-t, s-t y t-s entregan un menor error con 4 nodos ocultos, mientras la red neuronal s-s lo hace con 5 nodos, todas ellas entrenadas utilizando LDWPSO. Estos datos se grafican en la figura 6.1 en donde se aprecia claramente que la red que presentó menor error fue la red neuronal s-t entrenada mediante LDWPSO.

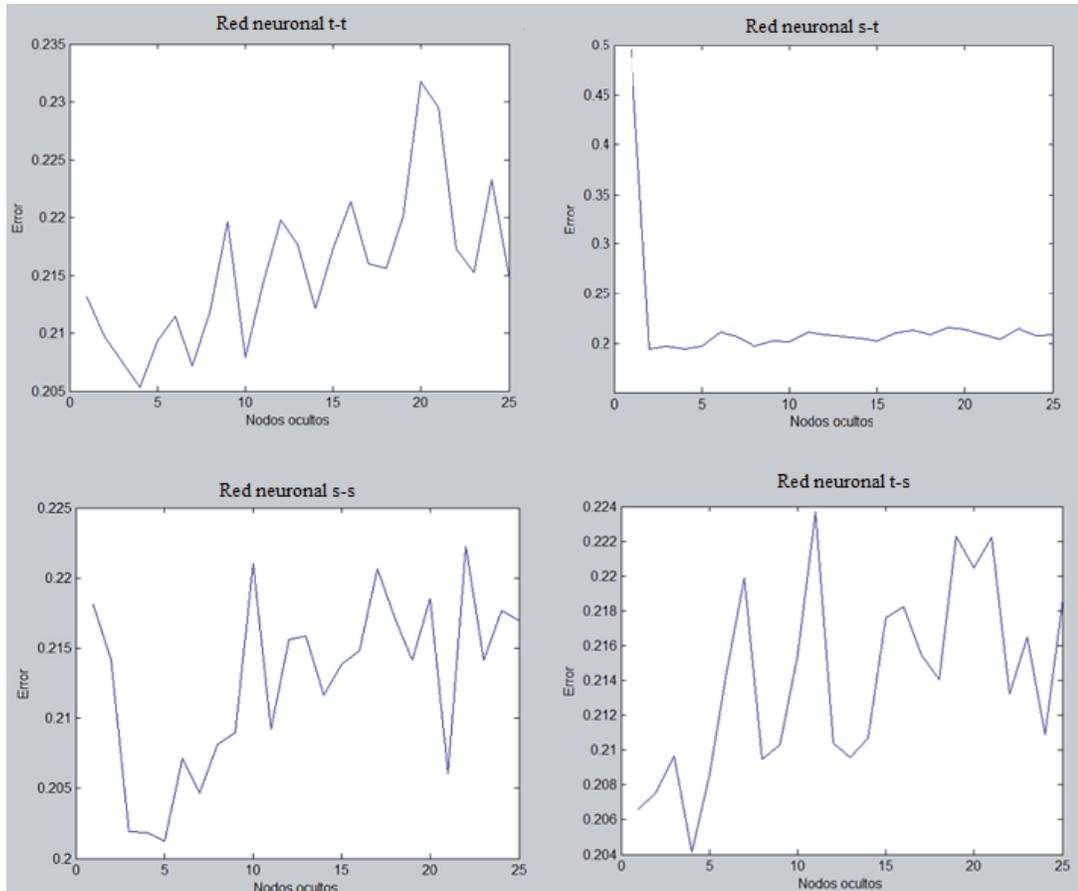


Figura 6.1 Error Promedio según el número de nodos ocultos para cada red propuesta, primeros modelos

### 6.1.3 Entrenar red neuronal mediante algoritmo LDWPSO y evaluar que red presenta mejores resultados.

Una vez se determinó el número de nodos ocultos para cada red propuesta, es necesario entrenar las redes para obtener pesos sinápticos que confieran a estas redes la posibilidad de obtener buenos resultados en cuanto a clasificación según sus parámetros de entrada.

Para entrenar las redes neuronales propuestas se utilizó el 80% de 3000 datos (el 20% restante se utilizará posteriormente para su evaluación), como algoritmo de entrenamiento se utilizó PSO linealmente decreciente (LDWPSO) con un factor de inercia que va desde 1 a 0.5, 500 iteraciones y 100 partículas. Esto se realizó 10 veces y se sacó el promedio para el error, la exactitud, la sensibilidad y la especificidad. Los resultados del entrenamiento para cada una de las redes propuestas pueden apreciarse en el anexo C en las tablas C.1, C.2, C.3 y C.4, así como también el resumen de estos en la tabla 6.2 en donde se aprecia claramente que el menor error, la mejor exactitud y la mejor especificidad las obtuvo la red neuronal s-t mientras que la mejor sensibilidad la obtuvo la red neuronal s-s.

**Tabla 6.2 Resultados promedio del entrenamiento, primeros modelos**

Funciones	No	Error	Exactitud	Especificidad	Sensibilidad
t-t	4	0,1943	0,8058	0,9178	0,6970
s-t	4	0,1888	0,8112	0,9420	0,6867
s-s	5	0,1914	0,8086	0,9197	0,6998
t-s	4	0,1900	0,8100	0,9477	0,6747

Una vez determinado los pesos de cada red se prosigue a evaluar. Los resultados de la evaluación para cada una de las redes puede apreciarse en las tablas D.1, D.2, D.3 y D.4, en ellas se valida un red 10 veces y se obtienen los valores máximos, los valores mínimos, la desviación estándar y el promedio del error para la exactitud, la sensibilidad y la especificidad.

De las tablas D.1, D.2, D.3 y D.4 se obtiene la tabla 6.3, la cual muestra el promedio para el error, la exactitud, la especificidad y la sensibilidad de cada red neuronal propuesta. En ella se puede apreciar que la red neuronal s-t presenta la mejor exactitud y especificidad, mas la mejor sensibilidad la presenta la red neuronal t-t.

**Tabla 6.3 Resultados promedio de la evaluación**

Funciones	No	Error	Exactitud	Especificidad	Sensibilidad
t-t	4	0,1933	0,8067	0,9087	0,7039
s-t	4	0,1930	0,8070	0,9371	0,6802
s-s	5	0,2055	0,7945	0,9078	0,6832
t-s	4	0,1998	0,8002	0,9436	0,6647

Otro aspecto importante es considerar la ganancia que presenta cada red neuronal propuesta sobre las demás, es por ello que se recoge la ecuación 5.1.2 de ganancia. En la tabla 6.4 se puede apreciar la ganancia de exactitud de cada red neuronal propuesta sobre las demás para los primeros modelos propuestos.

**Tabla 6.4 Ganancia de los resultados promedio de la evaluación**

A \ B	t-t	s-t	s-s	t-s
t-t	0%	-0,0372%	1,5123%	0,8058%
s-t	0,0372%	0%	1,5489%	0,8426%
s-s	-1,5356%	-1,5733%	0%	-0,7174%
t-s	-0,8123%	-0,8498%	0,7123%	0%

En la tabla 6.4 se puede apreciar que la única red que no tiene perdida de sino ganancia respecto a las demás es la red neuronal s-t, esto implica que es la red que presenta mejores resultados para clasificación entre las propuestas en los primeros modelos propuestos, basándose en la exactitud.

## 7. Segundos Modelos propuestos

La diferencia entre los segundos modelos propuestos y los primeros, es que en los segundos se utilizó validación cruzada (sección 3.3) y como función de desempeño en variantes de PSO se utilizó la fórmula 3.3.2 de entropía cruzada.

### 7.1 MLP

Para determinar los modelos se siguen los siguientes pasos una vez ya procesados los datos:

- Elegir las funciones de activación a utilizar.
- Determinar el número de nodos ocultos.
- Entrenar red neuronal mediante los algoritmos LDWPSO, QPSO y LDWQPSO a fin de evaluar que red presenta mejores resultados.

#### 7.1.1 Elegir las funciones de activación a utilizar

Se eligieron las funciones de activación sigmoideal y tangente hiperbólica formando dos redes neuronales:

- Red neuronal con función de activación tangente hiperbólica para la capa oculta y función sigmoideal para la capa de salida. Nos referirnos a esta red como **red neuronal t-s** a fin de facilitar su reconocimiento.
- Red neuronal con función de activación sigmoideal para capa oculta y de salida. Nos referirnos a ella como **red neuronal s-s** a fin de facilitar su reconocimiento.

#### 7.1.2 Determinar la Cantidad de Nodos Ocultos

Se busca determinar el número de nodos ocultos, para lo cual se utilizaron 3000 datos de los cuales se utilizó el 80% para obtener el desempeño durante el entrenamiento utilizando variantes de PSO con 500 iteraciones y 100 partículas para así realizar 10 pruebas con los datos de prueba, cada una de ellas desde 1 nodo oculto hasta 12 nodos ocultos dado que en los primeros modelos se detectó que después de ellos los resultados no mejoran, posteriormente se obtiene la media aritmética para así elegir el número de nodos ocultos que presente el menor error determinado por la fórmula 3.3.2 de ecuación de entropía cruzada.

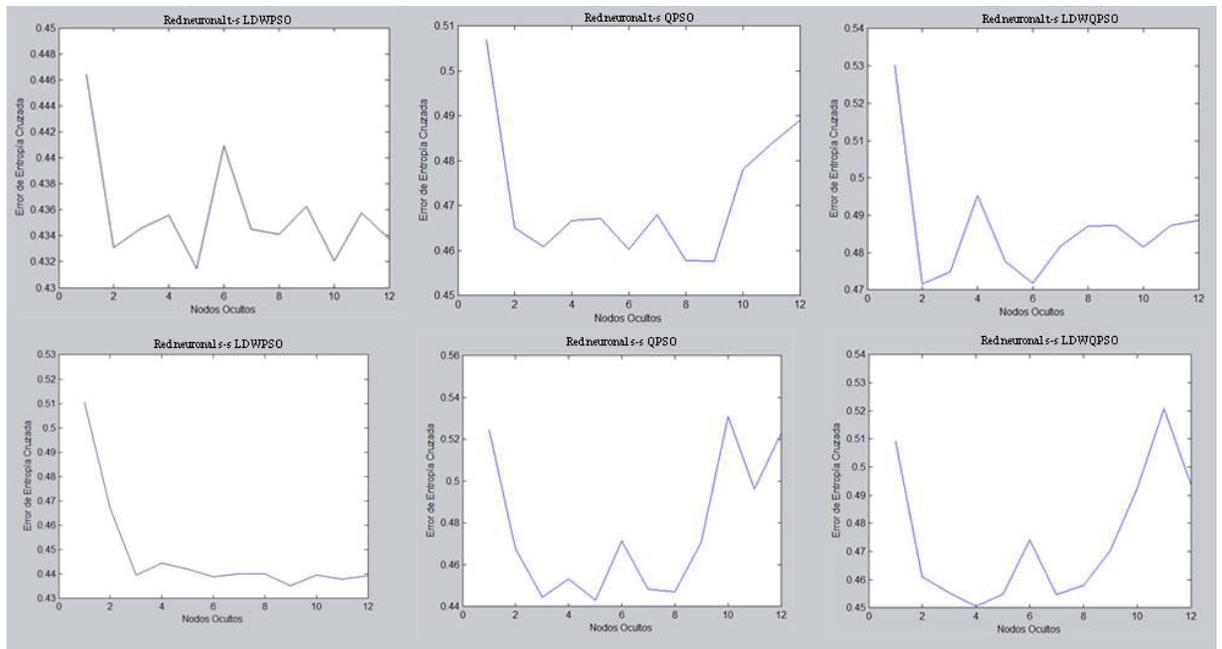
Se utilizaron tres tipos de algoritmos para entrenar cada una de las redes neuronales propuestas anteriormente, estos son: LDWPSO, QPSO y LDWQPSO. Los resultados para la obtención del número de nodos ocultos para cada red propuesta se pueden visualizar en la siguiente tabla, la cual muestra el número de nodos ocultos que presento el menor error promediado de diez pruebas realizadas.

**Tabla 7.1 Mejores resultados en elección de nodos ocultos, segundos modelos**

Red Neuronal	Algoritmo de Entrenamiento	N° de Nodos Ocultos	Max.	Min.	Desv.	Prom.
t-s	LDWPSO	5	0,4392	0,4241	0,0056	0,4315
t-s	QPSO	9	0,4780	0,4465	0,0098	0,4577
t-s	LDWQPSO	2	0,5098	0,4591	0,0148	0,4716
s-s	LDWPSO	9	0,4488	0,4290	0,0058	0,4351
s-s	QPSO	5	0,4656	0,4237	0,0119	0,4432
s-s	LDWQPSO	4	0,5054	0,4168	0,0347	0,4506

Los resultados de la tabla 7.1 muestran los mejores resultados de las tablas B.5 y B.6 perteneciente al anexo B, estos resultados pueden apreciarse en la figura 7.1, en ella se aprecia como varía el error de entropía cruzada promediado considerando desde un nodo oculto hasta doce para cada tipo de red propuesta en este capítulo.

Con los datos obtenidos, se puede determinar que las redes neuronal t-s y s-s entrenadas con LDWPSO y QPSO respectivamente entregan un error menor con 5 nodos ocultos, mientras que las redes neuronales t-s y s-s entrenadas con QPSO y LDWPSO repectivamente entregan un menor error con 9 nodos ocultos, por su parte la red neuronal s-s entrenada con LDWQPSO entrega un menor error con 4 nodos ocultos y la red neuronal t-s entrenda con LDWQPSO entrega un menor error con 2 nodos ocultos.



**Figura 7.1 Error de entropía cruzada promedio según el número de nodos ocultos para cada red propuesta, primeros modelos**

Los resultados que muestra la tabla 7.1 se pueden apreciar en la figura 7.1, en ella se aprecia como varía el error promediado considerando desde un nodo oculto hasta doce para cada tipo de red.

### 7.1.3 Entrenar y evaluar la red neuronal t-s mediante algoritmos LDWPSO, QPSO y LDWQPSO

Una vez se determinó el número de nodos ocultos para cada red propuesta, es necesario entrenar las redes para obtener los pesos sinápticos que confieran a estas redes la posibilidad de obtener buenos resultados en cuanto a clasificación según sus parámetros de entrada.

Para cada tipo de red se consideraron 4 muestras de datos de 1000, 3000, 10000 y 20000 datos con lo que se entrenaron 4 redes neuronales respectivamente. Para cada una se consideró un 80% de los datos y el resto se utilizó para evaluar los resultados.

Para entrenar las redes se utilizó la técnica de validación cruzada con un k de 10 junto con los algoritmos LDWPSO, QPSO y LDWQPSO, considerando un factor de inercia que fue desde 1 a 05 para LDWPSO y LDWQPSO, además se utilizaron 500 iteraciones y 100 partículas en todos los casos. Los resultados obtenidos de este entrenamiento pueden apreciarse en las tablas C.5, C.6 y C.7, en ellas se puede apreciar la variante de PSO que se utilizó, el tamaño de la muestra utilizada, el número de nodos ocultos, el error de entropía cruzada, la exactitud, la especificidad y la sensibilidad, así también la tabla 7.2 muestran los resultados de las muestras promediados para cada red propuesta (el promedio que se puede apreciar en las tablas C.5, C.6 y C.7) mostrando la variante de PSO que se utilizó, el número de nodos ocultos, el error de entropía cruzada, la exactitud, la especificidad y la sensibilidad.

**Tabla 7.2 Resultados promedios del entrenamiento de tablas C.5, C.6 y C.7, segundos modelos**

Tipo de PSO	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	5	0,4348	0,8060	0,9416	0,6785
QPSO	9	0,4616	0,7939	0,9408	0,6490
LDWQPSO	2	0,4696	0,7794	0,9421	0,6289

De la tabla 7.2 se puede rescatar que para el entrenamiento utilizando validación cruzada con un k de 10 en una red neuronal t-s, se tiene que: el menor error, la mayor exactitud y la mayor sensibilidad obtenida del promedio de los resultados de las muestras (apreciables en las tablas C.5, C.6 y C.7) se obtuvieron con LDWPSO y 5 nodos ocultos; en cambio la mayor especificidad obtenida del promedio de los resultados de las muestras (apreciables en las tablas C.5, C.6 y C.7) se obtuvo con LDWQPSO y 2 nodos ocultos.

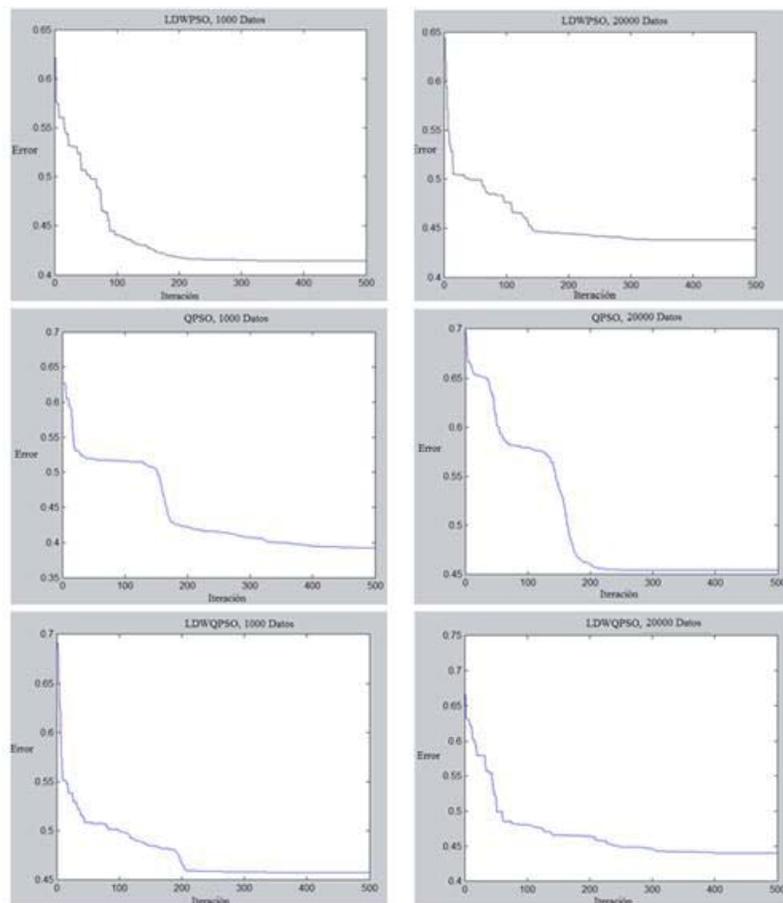
Una vez determinado los pesos de cada red se prosigue a evaluar. Los resultados de la evaluación para cada una de las redes con sus diversos algoritmos de entrenamiento, tamaños de muestra y el promedio de estos resultados se puede apreciar en las tablas D.5, D.6 y D.7 del anexo D, en ellas se obtiene el error de entropía cruzada, la exactitud, la especificidad y la

sensibilidad, promediando estas métricas obtenidas para cada muestra (de 1000, 3000, 10000 y 20000 datos).

De las tablas D.5, D.6 y D.7 del anexo D se obtiene la tabla 7.3, la cual muestra el tipo de algoritmo PSO utilizado, además del error, la exactitud, la especificidad y la sensibilidad promedio de las muestras de datos (1000, 3000, 10000, 20000) de cada red neuronal propuesta. En ella se puede apreciar que la red neuronal s-t con 9 nodos ocultos entrenada con QPSO presenta el menor error de entropía cruzada, la mejor exactitud y sensibilidad promedios, mientras que la mejor especificidad promedio la presenta la red neuronal s-t con 2 nodos ocultos entrenada con LDWQPSO.

**Tabla 7.3 Resultados promedios de la evaluación en tablas D.5, D.6 y D.7, segundos modelos**

Tipo de PSO	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	5	0,4288	0,8084	0,9393	0,6808
QPSO	9	0,4216	0,8111	0,9355	0,6953
LDWQPSO	2	0,4702	0,7922	0,9660	0,6167



**Figura 7.2 Convergencias de tipos de PSO**

Respecto a la convergencia de los diversos algoritmos (LDWPSO, QPSO y LDWQPSO) con diversos tamaños de muestras en la etapa de entrenamiento mediante validación cruzada (donde se guardaron los valores de convergencia que presentaron menor error de entropía cruzada) para los algoritmos se obtuvieron los gráficos expuestos en la figura 7.2.

Como se puede observar en la figura anterior, la convergencia bordea un error de 0.45. Visualmente se puede apreciar que LDWPSO muestra un descenso más homogéneo y no tan brusco como QPSO, mientras LDWQPSO muestra un descenso mezcla de los dos algoritmos nombrados anteriormente.

#### 7.1.4 Entrenar y evaluar la red neuronal s-s mediante algoritmos LDWPSO, QPSO y LDWQPSO

Ya habiendo determinado el número de nodos ocultos para cada red propuesta, se entrenan las redes para obtener pesos sinápticos que confieran a estas redes la posibilidad de obtener buenos resultados en cuanto a clasificación según sus parámetros de entrada.

Para cada tipo de red se consideraron 4 muestras de datos de 1000, 3000, 10000 y 20000 datos con lo que se entrenaron 4 redes neuronales respectivamente. Para cada una se consideró un 80% de los datos y el resto se utilizó para evaluar los resultados.

Para entrenar las redes se utilizó la técnica de validación cruzada con un k de 10 junto con los algoritmos LDWPSO, QPSO y LDWQPSO, considerando un factor de inercia desde 1 a 0,5 para LDWPSO y LDWQPSO, además se utilizaron 500 iteraciones y 100 partículas en todos los casos. Los resultados obtenidos de este entrenamiento pueden apreciarse en las tablas C.8, C.9 y C.10, en ellas se puede apreciar la variante de PSO que se utilizó, el tamaño de la muestra utilizada, el número de nodos ocultos, el error de entropía cruzada, la exactitud, la especificidad y la sensibilidad, así también la tabla 7.4 muestran los resultados de las muestras promediadas para cada red propuesta (el promedio que se puede apreciar en las tablas C.8, C.9 y C.10) mostrando la variante de PSO que se utilizó, el número de nodos ocultos, el error de entropía cruzada, la exactitud, la especificidad y la sensibilidad.

**Tabla 7.4 Resultados promedios del entrenamiento de tablas C.5, C.6 y C.7, segundos modelos**

Tipo de PSO	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	9	0,4438	0,8006	0,9328	0,6730
QPSO	5	0,4513	0,7916	0,9152	0,6684
LDWQPSO	4	0,4650	0,7802	0,9119	0,6553

De la tabla 7.4 se puede rescatar que para el entrenamiento utilizando validación cruzada con un k de 10 en una red neuronal t-s, se tiene que: el menor error, la mayor exactitud, la mayor especificidad y la mayor sensibilidad obtenida del promedio de los resultados de las muestras (apreciables en las tablas C.8, C.9 y C.10) se obtuvieron con LDWPSO y 9 nodos ocultos.

Una vez determinado los pesos de cada red se prosiguió a evaluar. Los resultados de la evaluación para cada una de las redes con sus diversos algoritmos de entrenamiento, y tamaños de muestra y el promedio de esos resultados pueden observarse en las tablas D.8, D.9 y D.10 del anexo D, en ellas se obtiene el error de entropía cruzada, la exactitud, la especificidad y la sensibilidad, promediando estas métricas obtenidas en cada muestra (de 1000, 3000, 10000 y 20000 datos).

De las tablas D.8, D.9 y D.10 del anexo D se obtiene la tabla 7.3, la cual muestra el tipo de variante de PSO utilizada, además del error, la exactitud, la especificidad y la sensibilidad promedio para las muestras de datos (1000, 3000, 10000, 20000) de cada red neuronal propuesta. En ella se puede apreciar que la red neuronal s-s con 4 nodos ocultos entrenada con LDWQPSO presenta el menor error de entropía cruzada y la mejor especificidad, mientras que la red neuronal s-s con 5 nodos ocultos entrenada con QPSO presenta la mejor exactitud y sensibilidad.

**Tabla 7.5 Resultados promedios de la evaluación en tablas D.8, D.9 y D.10, segundos modelos**

Tipo de PSO	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	9	0,4552	0,7874	0,9093	0,6739
QPSO	5	0,4837	0,7992	0,9042	0,7000
LDWQPSO	4	0,4524	0,7975	0,9487	0,6508

Respecto a la convergencia de los diversos algoritmos (LDWPSO, QPSO y LDWQPSO) con diversos tamaños de muestras en la etapa de entrenamiento mediante validación cruzada (donde se guardaron los valores de convergencia que presentaron menor error de entropía cruzada) para los algoritmos se obtuvieron los gráficos expuestos en la figura 7.3.

Como se puede observar en la figura 7.3, con el algoritmo LDWPSO la convergencia se logra alrededor de la iteración 300 con un error de entropía cruzada que bordea 0.45, comparado con QPSO y LDWQPSO, LDWPSO presenta el menor error obtenido, le sigue QPSO el cual tiene un error que bordea el valor de 0,62 y después LDWQPSO que tiene un error que bordea los 0,63. QPSO converge antes que LDWQPSO para una muestra de 20000 datos, además de tener un error menor. Por ende en cuanto a convergencia LDWPSO presenta mejores resultados, después le sigue QPSO y posteriormente LDWQPSO.

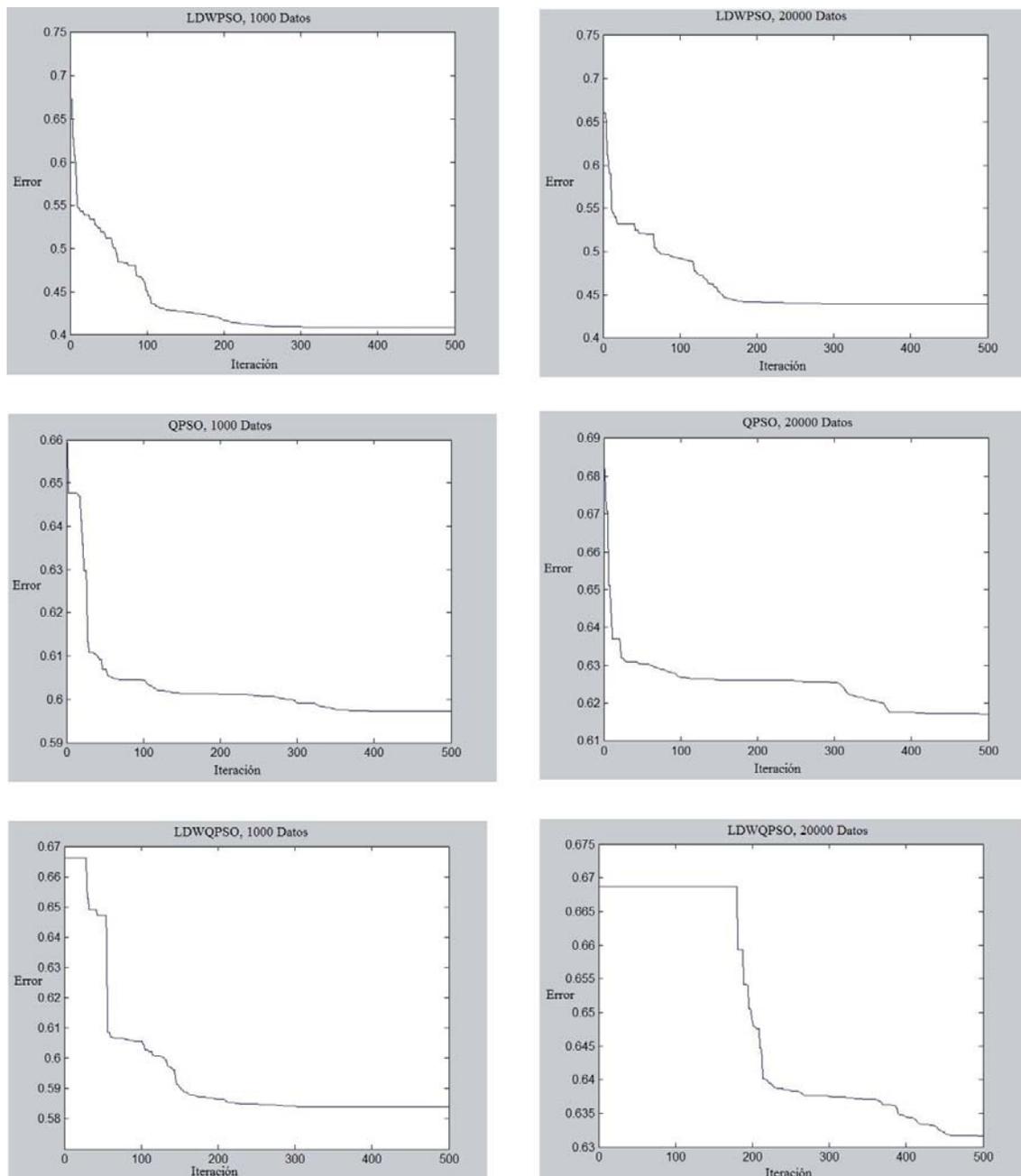


Figura 7.3 Convergencias de tipos de PSO

### 7.1.5 Comparación de resultados

Los resultados ofrecidos por las tablas anteriores pueden ser resumidos en la tabla 7.6 mostrando el promedio para: el error, la exactitud, la especificidad y la sensibilidad de cada red neuronal propuesta. En la tabla 7.6 se puede apreciar que la red neuronal t-s con algoritmo QPSO para su entrenamiento y con 9 nodos ocultos presenta el menor error y la mejor exactitud promedios. La mejor especificidad está dada por la red neuronal t-s con algoritmo LDWQPSO para su entrenamiento y con 2 nodos ocultos, mientras que la mejor sensibilidad

está dada por la red neuronal s-s con el algoritmo QPSO para su entrenamiento y con 5 nodos ocultos.

**Tabla 7.6 Resultados promedio de la evaluación**

Funciones	Algoritmo	No	Error	Exactitud	Especificidad	Sensibilidad
t-s	LDWPSO	5	0,4288	0,8084	0,9393	0,6808
t-s	QPSO	9	0,4216	0,8111	0,9355	0,6953
t-s	LDWQPSO	2	0,4702	0,7922	0,9660	0,6167
s-s	LDWPSO	9	0,4552	0,7874	0,9093	0,6739
s-s	QPSO	5	0,4837	0,7992	0,9042	0,7000
s-s	LDWQPSO	4	0,4524	0,7975	0,9487	0,6508

Otro aspecto importante es considerar la ganancia que representa cada red neuronal propuesta sobre las demás, es por ello que se recurre a la fórmula de ganancia. En la tabla 7.7 se puede apreciar la ganancia de exactitud de cada red neuronal propuesta sobre las demás.

**Tabla 7.7 Ganancia de los resultados promedio de la evaluación**

A \ B	t-s-LDWPSO	t-s-QPSO	t-s-LDWQPSO	s-s-LDWPSO	s-s-QPSO	s-s-LDWQPSO
t-s-LDWPSO	0%	-0,3340%	2,0040%	2,5977%	1,1381%	1,3483%
t-s-QPSO	0,3329%	0%	2,3302%	2,9220%	1,4671%	1,6767%
t-s-LDWQPSO	-2,0449%	-2,3858%	0%	0,6059%	-0,8836%	-0,6690%
s-s-LDWPSO	-2,6670%	-3,0099%	-0,6096%	0%	-1,4986%	-1,2827%
s-s-QPSO	-1,1512%	-1,4890%	0,8759%	1,4765%	0%	0,2127%
s-s-LDWQPSO	-1,3668%	-1,7053%	0,6646%	1,2665%	0,6646%	0%

En la tabla 7.7 se puede apreciar que la única red que no tiene pérdida de sino ganancia respecto a las demás es la red neuronal t-s entrenada mediante QPSO, esto implica que es la red que presenta mejores resultados para clasificación entre las propuestas en los segundos modelos.

## 8. Terceros Modelos propuestos

La diferencia con los modelos realizados anteriormente se encuentra en que la red neuronal artificial considerada en los terceros modelos considera dos salidas, una para lesionados y otra para no lesionados, a diferencia de las anteriores que solo consideraban una neurona de salida para determinar tanto lesionados como ilesos, además se ocupa como función de salida la función *softmax* para poder determinar a qué clase pertenece la salida. Otro punto a considerar es el hecho de que como función de desempeño en las variantes de PSO se utilizó la fórmula 3.3.3 de entropía cruzada en lugar de la fórmula 3.3.2. Por otro lado se utiliza la técnica de validación cruzada.

### 8.1 MLP

Para determinar los modelos se siguieron los siguientes pasos una vez ya procesados los datos:

- Elegir las funciones de activación a utilizar.
- Determinar el número de nodos ocultos.
- Entrenar red neuronal mediante algoritmo LDWPSO, QPSO y LDWQPSO a fin de evaluar que red presenta mejores resultados.

#### 8.1.1 Elegir las funciones de activación a utilizar

Se eligió la función sigmoideal para los nodos de la capa oculta y la función softmax para la capa de salida, de esta manera es posible obtener el resultado mediante uno o más nodos de salida, para el caso se busca determinar el resultado mediante dos nodos de salida y así poder clasificar según los atributos de entrada si es que la persona esta ilesa o lesionada.

A la red neuronal con función de activación sigmoideal para la capa oculta y softmax para capa de salida se le llamará **red neuronal sigmoideal-softmax**.

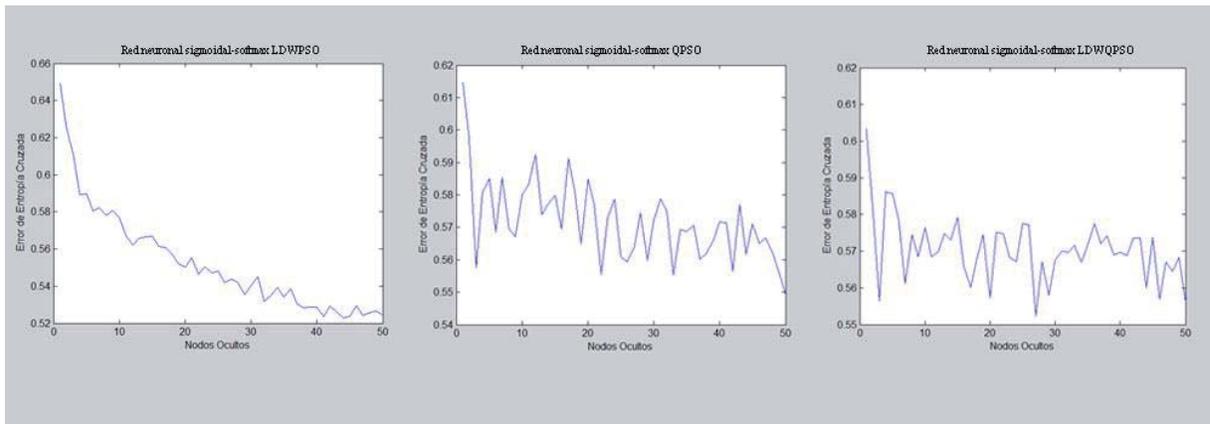
#### 8.1.2 Determinar la Cantidad de Nodos Ocultos

Se busca determinar el número de nodos ocultos, para lo cual se utilizaron 3000 datos de los cuales se utilizó el 80% para el desempeño del entrenamiento con 500 iteraciones y 100 partículas para así realizar 10 pruebas con los datos de prueba, cada una de ellas desde 1 nodo oculto hasta 50 nodos ocultos dado que a diferencia de los segundos modelos se desconoce si mejorará el rendimiento después o antes de 25 nodos ocultos, posteriormente se obtiene la media aritmética para así elegir el número de nodos ocultos que presente el menor error determinado por la fórmula 3.3.3 de entropía cruzada.

**Tabla 8.1 Mejores resultados en elección de nodos ocultos, terceros modelos**

Red Neuronal	Algoritmo de Entrenamiento	Nº de Nodos Ocultos	Max.	Min.	Desv.	Prom.
<b>sigmoidal-softmax</b>	LDWPSO	44	0,5423	0,5072	0,0114	0,5227
<b>sigmoidal-softmax</b>	QPSO	50	0,5794	0,5081	0,0261	0,5493
<b>sigmoidal-softmax</b>	LDWQPSO	27	0,5977	0,5096	0,0329	0,5523

Se utilizaron tres tipos de algoritmos para entrenar la red neuronal propuesta anteriormente, estos son: LDWPSO, QPSO y LDWQPSO. Los resultados para la obtención del número de nodos ocultos para cada red propuesta se pueden visualizar en la tabla 8.1, la cual muestra el número de nodos ocultos que presentó el menor error promediado de diez pruebas realizadas. Estos resultados son obtenidos de la tabla B.1 del anexo B, la cual muestra los diversos resultados obtenidos desde 1 nodo hasta 50, estos resultados son graficados y mostrados en la figura 8.1, en ella se aprecia claramente y que la variación más suave se obtiene utilizando LDWPSO, así también se aprecia una razón inversamente proporcional al número de nodos ocultos (dentro del rango considerado), aun así por razones de tiempo no se ahondo más allá de los 50 nodos ocultos.



**Figura 8.1 Error de entropía cruzada promedio según el número de nodos ocultos para cada red propuesta, segundos modelos**

### 8.1.3 Entrenar y evaluar la red neuronal sigmoidal-softmax mediante algoritmos LDWPSO, QPSO y LDWQPSO

Una vez determinadas las cantidades de nodos ocultos se procede a probar las redes con diversos tamaños de muestras y algoritmos PSO. Considerando 12 nodos de entrada y a diferencia de los modelos anteriores, en estos modelos se consideran 2 nodos de salida.

Se consideraron 4 muestras de datos de 1000, 3000, 10000 y 20000 datos. Para cada una se consideró un 80% de los datos para entrenamiento y el resto se utilizó para evaluar los resultados.

Para entrenar las redes se utilizó la técnica de validación cruzada con un k de 10 junto con los algoritmos LDWPSO, QPSO y LDWQPSO, considerando un factor de inercia que fue desde 1 a 05 para LDWPSO y LDWQPSO, además se utilizaron 500 iteraciones y 100 partículas en todos los casos. Los resultados obtenidos de este entrenamiento pueden apreciarse en las tablas C.11, C.12 y C.13, en ellas se puede apreciar la variante de PSO que se utilizó, el tamaño de la muestra utilizada, el número de nodos ocultos, el error de entropía cruzada, la exactitud, la especificidad y la sensibilidad, así también la tabla 8.2 muestran los resultados de las muestras promediados para cada red propuesta (el promedio que se puede apreciar en las tablas C.11, C.12 y C.13) mostrando la variante de PSO que se utilizó, el número de nodos ocultos, el error de entropía cruzada, la exactitud, la especificidad y la sensibilidad.

**Tabla 8.2 Resultados promedios del entrenamiento de tablas C.11, C.12 y C.13, terceros modelos**

Tipo de PSO	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	44	0,4963	0,8170	0,8883	0,7441
QPSO	50	0,5467	0,7666	0,8462	0,6841
LDWQPSO	27	0,5417	0,7716	0,8434	0,6976

De la tabla 8.2 se puede rescatar que para el entrenamiento donde se utilizó validación cruzada con un k de 10 en una red neuronal sigmooidal-softmax, se tiene que: el menor error, la mayor exactitud, la mayor sensibilidad y la mayor especificidad obtenida del promedio de los resultados de las muestras (apreciables en las tablas C.11, C.12 y C.13) se obtuvieron con LDWPSO y 44 nodos ocultos.

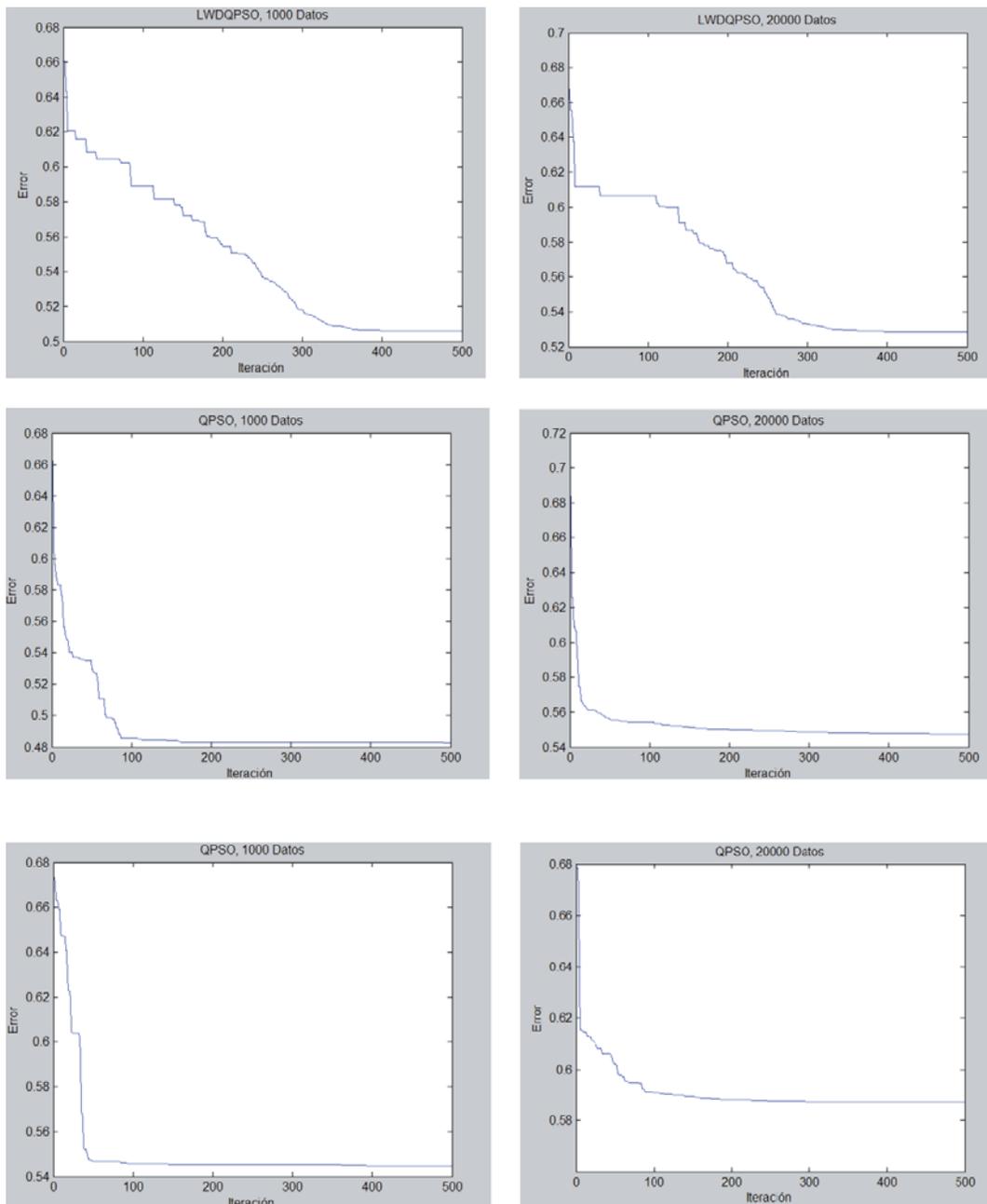
Una vez determinado los pesos de cada red se prosigue a evaluar. Los resultados de la evaluación para cada una de las redes con sus diversos algoritmos de entrenamiento, y tamaños de muestras, promediando estos resultados como puede apreciarse en las tablas D.11, D.12 y D.13 del anexo D, en ellas se obtiene el error de entropía cruzada, la exactitud, la especificidad y la sensibilidad, promediando estas métricas obtenidas en cada muestra (de 1000, 3000, 10000 y 20000 datos).

De las tablas D.11, D.12 y D.13 del anexo D se obtiene la tabla 8.3, la cual muestra la variante de PSO utilizada, además del error, la exactitud, la especificidad y la sensibilidad promedio de las muestras de datos (1000, 3000, 10000, 20000) de cada red neuronal propuesta. En ella se puede apreciar que la red neuronal sigmooidal-softmax con 50 nodos ocultos entrenada con QPSO presenta el menor error de entropía cruzada, la mejor exactitud y sensibilidad promedios, mientras que la mejor especificidad promedio la presenta la red neuronal sigmooidal-softmax con 44 nodos ocultos entrenada con LDWPSO.

**Tabla 8.3 Resultados promedios de la evaluación en tablas D.11, D.12 y D.13, terceros modelos**

Tipo de PSO	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	44	0,5440	0,7693	0,8609	0,6767
QPSO	50	0,5374	0,7759	0,8316	0,7122
LDWQPSO	27	0,5610	0,7523	0,8492	0,6614

Respecto a la convergencia de los diversos algoritmos (LDWPSO, QPSO y LDWQPSO) con diversos tamaños de muestras en la etapa de entrenamiento mediante validación cruzada (donde se guardaron los valores de convergencia que presentaron menor error de entropía cruzada) para los algoritmos se obtuvieron los gráficos expuestos en la figura 8.2.



**Figura 8.2** Convergencias de tipos de PSO

Como se puede observar en la figura 8.2, con el algoritmo LDWPSO la convergencia es más gradual a diferencia de QPSO y LDWQPSO en donde el error cae rápidamente hasta lograr cierto grado de estabilidad presentando pequeñas variaciones en el error. Con una

muestra de 1000 datos, se puede observar que el menor error se logra con QPSO con un error cercano a 0.48. Si bien el error deja de fluctuar alrededor de la iteración 500, este se empieza a estabilizar cerca de la iteración 200, para LDWQPSO y es similar, mientras que para LDWPSO la convergencia es cercana a la iteración 400 con un descenso más gradual. Para una muestra de 20000 el menor error se logra con LDWPSO logrando una convergencia cerca de la iteración 400, a diferencia de QPSO y LDWQPSO, la convergencia es más gradual, para QPSO la convergencia es cercana a la iteración 500, mientras que para LDWQPSO esta se logra cerca de la iteración 300.

La figura 8.2 deja ver además que la convergencia tiene un error cada vez más alto a medida que aumenta el número de la muestra.

### 8.1.4 Comparación de resultados

Como se puede apreciar en la tabla 8.3, la red neuronal sigmooidal-softmax con 50 nodos ocultos entrenada con QPSO presenta el menor error de entropía cruzada, la mejor exactitud y sensibilidad promedios, mientras que la mejor especificidad promedio la presenta la red neuronal sigmooidal-softmax con 44 nodos ocultos entrenada con LDWPSO.

Desde otra perspectiva, es importante considerar la ganancia que representa cada red neuronal propuesta sobre las demás, es por ello que se recurre a la fórmula de ganancia. En la siguiente tabla se puede apreciar la ganancia de exactitud de cada red neuronal propuesta sobre las demás.

**Tabla 8.4 Ganancia de los resultados promedio de la evaluación**

A \ B	sigmooidal-softmax - LDWPSO	sigmooidal-softmax - QPSO	sigmooidal-softmax - LDWQPSO
sigmooidal-softmax - LDWPSO	0%	-0,8579%	2,2098%
sigmooidal-softmax - QPSO	0,8506%	0%	3,0416%
sigmooidal-softmax - LDWQPSO	-2,2597%	-3,1371%	0%

En ella se puede apreciar que la única red que no tiene pérdida de sino ganancia respecto a las demás es la red neuronal sigmooidal-softmax entrenada mediante QPSO, esto implica que es la red que presenta mejores resultados para clasificación entre las propuestas en los terceros modelos propuestos.

## 9. Comparación de resultados

Si bien los segundos y terceros modelos propuestos ocuparon más técnicas de PSO y otras métricas para calcular el error, es posible compara los resultados para muestras de 3000 datos, entregados tanto por los primeros como por los segundos y tercero modelos propuestos.

Como se aprecia en la tabla 6.4 el modelo que presenta una ganancia por sobre todos los demás de los primeros modelos vistos es el que posee una red neuronal con 12 nodos de entrada, 4 nodos de capa oculta y 1 de salida, siendo sus funciones de activación sigmoideal para la capa oculta y tangente hiperbólica para la capa de salida. Al comparar esta red con los resultados obtenidos para los demás modelos propuestos (segundos y terceros) para una muestra de 3000 datos, es que se obtiene la siguiente tabla , siendo una compilación de los resultados expuestos en las tablas D.5, D.6, D.7, D.8, D.9, D.10, D.11, D.12, D.13 , para 3000 datos de muestra.

**Tabla 9.1 Comparación de modelos con una muestra de 3000 datos**

Modelos	PSO	Funciones	No	Exactitud	Especificidad	Sensibilidad
Primeros	LDWPSO	sigmoideal-tangente hiperbólica	4	0,8070	0,9371	0,6802
segundos	LDWPSO	tangente hiperbólica-sigmoideal	5	0,8017	0,9519	0,6602
segundos	QPSO	tangente hiperbólica-sigmoideal	9	0,7950	0,9634	0,6544
segundos	LDWQPSO	tangente hiperbólica-sigmoideal	2	0,7717	0,9492	0,6000
segundos	LDWPSO	Sigmoideal-sigmoideal	9	0,8233	0,9483	0,7065
segundos	QPSO	Sigmoideal-sigmoideal	5	0,8200	0,9519	0,6958
segundos	LDWQPSO	Sigmoideal-sigmoideal	4	0,7850	0,9524	0,6242
terceros	LDWPSO	Sigmoideal-softmax	44	0,7650	0,9123	0,6096
terceros	QPSO	Sigmoideal-softmax	50	0,7667	0,8520	0,6791
terceros	LDWQPSO	Sigmoideal-softmax	27	0,7083	0,7716	0,6495

Como se puede apreciar en la tabla 9.1, para una muestra de 3000 datos, al comparar la exactitud, se puede observar claramente que la red neuronal sigmoideal sigmoideal de los segundos modelos entrenada mediante LDWPSO con 9 nodos ocultos y un nodo de salida presenta la mejor exactitud y sensibilidad, mientras que la mejor especificidad se obtiene mediante la red neuronal tangente hiperbólica-sigmoideal de los segundos modelos entrenada con QPSO con 9 nodos ocultos y un nodo de salida.

Dado que los segundos y terceros modelos consideran diversos tamaños de conjuntos para entrenar y testear es que se comparan los resultados de sus evaluaciones con los resultados obtenidos en [4] por Felix Castro en la tabla 9.2.

**Tabla 9.2 Resultados obtenidos de [4] contrastados con segundos y terceros modelos**

	LS-SVM			Tangente hiperbólica-Sigmoidal			Sigmoidal-Sigmoidal			Sigmoidal - <i>Softmax</i>		
	PSO F.I.L	IPSO	DAPSO	LDWPSO	QPSO	LDWQPSO	LDWPSO	QPSO	LDWQPSO	LDWPSO	QPSO	LDWQPSO
1000	0,8590	0,845	0,848	0,8150	0,8350	0,79,50	0,7300	0,7950	0,8,50	0,7400	0,7500	0,7450
3000	0,837	0,8369	0,842	0,8017	0,7950	0,7717	0,8233	0,8200	0,7,50	0,7650	0,7667	0,7083
10000	0,818	0,7958	0,8145	0,8100	0,80,65	0,8135	0,7860	0,7945	0,7,90	0,7725	0,7895	0,7715
20000	0,8009	0,7992	0,8026	0,8070	0,80,77	0,7887	0,8103	0,7873	0,8,10	0,7998	0,7973	0,7845
Prom	0,8,87	0,8192	0,8268	0,8084	0,81,11	0,7922	0,7874	0,7992	0,7975	0,7693	0,7759	0,7523

De la tabla 9.2 se puede observar que los resultados relativos a la exactitud promediada de las muestras en las técnicas de SVM propuestas en [4] obtuvieron mejores resultados que los modelos propuestos en este trabajos, mas si se analizan los resultados presentados por cada muestra de datos es posible rescatar que:

- En las redes neuronales propuestas los resultados no guardan la fuerte relación que existe en [4] respecto al tamaño maño de la muestras en relación a la exactitud mostrada, lo que se puede apreciar en [4], en donde se aprecia claramente que a mayor tamaño de la muestra, más decaen los resultados del modelo SVM.
- Mientras que para muestras inferiores o iguales a 10000 datos la exactitud mayor siempre la posee alguno de los modelos presentados en [4], para una muestra de 20000 datos hay redes neuronales propuestas por en los segundos modelos que superan la exactitud obtenida por los modelos presentados en [4] cómo es posible visualizar en la tabla 9.2.

Si se comparan los resultados promediados de las muestras que presentaron mejor exactitud porcentual para el resultado del presente trabajo con los de [4] y el realizado en la Escuela en Transportes de la PUCV y miembros de la Universidad USACH [5], en este último utilizando redes bayesianas y como métricas la exactitud, sensibilidad y especificidad, se puede apreciar la siguiente tabla:

**Tabla 9.3 LS-SVM PSO vs Redes Bayesianas vs RNA QPSO**

	LS-SVM PSO F.I.L	Redes Bayesianas	Red neuronal artificial Tangente Hiperbólica Sigmoidal QPSO
Sensibilidad	78,12%	71%	69,53%
Especificidad	87,01%	87%	93,55%
Exactitud	82,04%	78%	81,11%

En la tabla 9.3 queda claro que el modelo propuesto supera en más de tres puntos al de redes bayesianas, mas no así en comparación al modelo LS-SVM PSO F.I.L, el cual saca casi un punto más de exactitud.

Es importante señalar que estas comparaciones son realizadas en base al promedio de distintas muestras de los resultados obtenidos con QPSO, no así en los trabajos relacionados que no utilizan un promedio de varias muestras, sino más bien, el mejor resultado para una muestra en particular. Por ende, las diferencias podrían ser aún más elevadas con el modelo obtenido en este trabajo de investigación.

Como se puede observar en la tabla 9.3, respecto a sensibilidad el modelo LS-SVM entrenado con PSO F.I.L presenta los mejores resultados seguidos de Redes Bayesianas, siendo la red neuronal artificial tangente hiperbólica sigmoïdal entrenada con QPSO la que presenta el resultado más bajo. Respecto a la especificidad, el mejor resultado lo presenta la red neuronal artificial tangente hiperbólica sigmoïdal entrenada con QPSO, seguida de LS-SVM entrenado con PSO F.I.L y seguido de las redes bayesianas. En cuanto a Exactitud, el mejor resultado lo presenta LS-SVM entrenado con PSO F.I.L, seguido de la red neuronal artificial tangente hiperbólica sigmoïdal entrenada con QPSO y seguida de redes bayesianas.

Si analizamos la exactitud mostrada por los mejores modelos desde el punto de vista de la exactitud, respecto a la ganancia obtenida al comparar la exactitud de los modelos, se puede rescatar según la tabla 9.4, que:

- El modelo que presenta ganancia positiva en todas las comparaciones es LS-SVM entrenado con PSO F.I.L.
- La red neuronal artificial Tangente Hiperbólica Sigmoïdal entrenada con QPSO presenta ganancia positiva respecto las redes bayesianas, no así respecto a LS-SVM entrenado con PSO F.I.L.
- Las redes Bayesianas presentan ganancia negativa frente a los otros modelos planteados en la tabla 9.4.

**Tabla 9.4 Ganancia de los resultados promedio de la evaluación de los modelos de la tabla 9.3**

A \ B	LS-SVM PSO F.I.L	Redes Bayesianas	Red neuronal artificial Tangente Hiperbólica Sigmoïdal QPSO
LS-SVM PSO F.I.L	0%	4,9244%	1,1336%
Redes Bayesianas	-5,1795%	0%	-3,9851%
Red neuronal artificial Tangente Hiperbólica Sigmoïdal QPSO	-1,1466%	3,8343%	0%

## 10. Conclusión

A través del desarrollo del presente proyecto se ha logrado entender el funcionamiento base de las redes neuronales artificiales tratadas, pudiéndose razonar sus principales ventajas y desventajas, lo que las hace una poderosa herramienta, la cual puede seguir evolucionando. También se pudo rescatar las diferencias entre las redes expuestas, en donde destaca como principal diferencia su función de activación.

Se logró una comprensión del algoritmo PSO y de algunas de sus variantes, con lo cual se logró dar cuenta de su simplicidad y eficiencia, razones que permiten justificar la elección de este algoritmo evolutivo como medio para el entrenamiento de las redes neuronales artificiales.

También se ha expuesto la problemática a resolver enmarcando los límites del estudio a realizar y las técnicas a utilizar.

Asimismo, se logró el cumplimiento de los objetivos y las actividades propuestas a lo largo del desarrollo del trabajo.

Las técnicas propuestas demostraron ser robustas y tener buenos resultados los que fueron contrastados y comparados con otros trabajos relacionados.

Al comparar los mejores resultados de los tres trabajos expuestos:

1. “Clasificación de Accidentes de Tránsito en la Zona de Valparaíso Utilizando Redes Neuronales con Algoritmos PSO”, utilizando las funciones de activación tangente hiperbólica y sigmoideal para las capas oculta y de salida respectivamente. Entrenada con QPSO.
2. [4] LS-SVM basado en Optimización por Enjambres de Partículas para Clasificación de Accidentes de Tránsito. Entrenada con PSO F.I.L.
3. [5] Estructura Bayesianas en Accidentes de Tránsito.

La mejor exactitud, es decir el modelo que logró la mejor clasificación, se obtuvo en [4] seguido por “Clasificación de Accidentes de Tránsito en la Zona de Valparaíso Utilizando Redes Neuronales con Algoritmos PSO” y por último “Estructura Bayesianas en Accidentes de Tránsito”. Otra forma de ver estos resultados es mediante la ganancia, la cual explica si se ganó o se perdió (en caso de ser negativa) respecto a una métrica, para ello se utilizó la ecuación 5.1.2.

La ganancia de exactitud de [4] respecto a “Clasificación de Accidentes de Tránsito en la Zona de Valparaíso Utilizando Redes Neuronales con Algoritmos PSO” es de 1,13%.

La ganancia de exactitud de [4] respecto a [5] es de 4,92.

La mejor especificidad, es decir el modelo que clasifica mejor a las personas que resultan ilesas, se obtuvo “Clasificación de Accidentes de Tránsito en la Zona de Valparaíso Utilizando Redes Neuronales con Algoritmos PSO”, seguida de [4] y por último [47].

La ganancia de especificidad de “Clasificación de Accidentes de Tránsito en la Zona de Valparaíso Utilizando Redes Neuronales con Algoritmos PSO” respecto a [4] es de 6,99%.

La ganancia de especificidad de “Clasificación de Accidentes de Tránsito en la Zona de Valparaíso Utilizando Redes Neuronales con Algoritmos PSO” respecto a [5] es de 7,00%.

La mejor sensibilidad, es decir el modelo que clasifica mejor a las personas que resultan lesionadas, se obtuvo en [4], seguida de [47] y por último “Clasificación de Accidentes de Tránsito en la Zona de Valparaíso Utilizando Redes Neuronales con Algoritmos PSO”.

La ganancia de sensibilidad de [4] respecto a [5] es de 9,11%.

La ganancia de sensibilidad de [4] respecto a “Clasificación de Accidentes de Tránsito en la Zona de Valparaíso Utilizando Redes Neuronales con Algoritmos PSO” es de 11,00%.

Si bien el trabajo realizado presento diversas técnicas enfocadas en las redes neuronales artificiales, se pueden abordar muchas más, así como mejorar la forma de elegir el número de nodos de entrada y ocultos. Es así como el tema elegido permite ahondar más en futuros trabajos permitiendo la mejora de la técnica de clasificación, ya sea con diversas técnicas y/o métodos para entrenar estas.

## 11. Bibliografía

- [1] B. Crawford Labrín y N. Guerra Álvarez, *Optimización de funciones a través de Optimización por Enjambre de Partículas y Algoritmos Genéticos*.
- [2] «CONASET, COMISIÓN NACIONAL DE SEGURIDAD DE TRÁNSITO,» 2010. [En línea]. Available: [http://www.conaset.cl/portal/portal/default/estadisticas\\_generales](http://www.conaset.cl/portal/portal/default/estadisticas_generales).
- [3] S. Mohaghegi, Y. del Valle, G. Venayagamoorthy y R. Harley, «COMPARISON OF PSO AND BACKPROPAGATION FOR TRAINING RBF NEURAL NETWORKS FOR IDENTIFICATION OF A POWER SYSTEM WITH STATCOM,» [En línea]. Available: <http://web.mst.edu/~ganeshv/>.
- [4] F. Castro, *LS-SVM basado en Optimización por Enjambres de Partículas para Clasificación de Accidentes de Tránsito, Valparaíso, 2010*.
- [5] C. Montt, A. Zúñiga y M. Chacón, *Estructura Bayesianas en Accidentes de Tránsito, 2009*.
- [6] W. McCulloch y W. Pitts, 1943.
- [7] D. Hebb, *Organization of Behavior, Jphn Wiley & Sons, 1949*.
- [8] M. Minsky, *Neural-analog network and the brain model problem. PhD Thesis, Princeton University, 1954*.
- [9] F. Rosenblatt, *The perceptron: A perceiving and recognizing automation. Technical Report, Cornell Aeronautical Laboratory, 1957*.
- [10] F. Rosenblatt, *Principles of Neurodynamics, Washington: Spartan Books, 1962*.
- [11] B. Widrow, *Adaptative samplet-data systems - a statistical theory of adaption, Vol. 4, Welscon, 1959*.
- [12] M. Minsky y S. Paper, *Perceptrons, 1969*.
- [13] H. T. Abdelwahab y M. A. Abdel-Aty, *Development of Artificial Neural Network Models to Predict Driver injury Severity in Traffic Accidents at Signalized Intersections*.
- [14] L. Mussone, A. Ferrari y M. Oneta, «An analysis of urban collisions using an artificial intelligence model,» *Accident Analysis and Prevention, Milán, 1999*.
- [15] S. Y. Sohn y S. H. Lee, *Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea, 2002*.
- [16] M. M. Chong, A. Abraham y M. Paprzycki, *Traffic Accident Analysis Using Decision Trees and Neural Network, Oklahoma State University, 2004*.
- [17] J. Zúñiga, *Aplicación de Redes Bayesianas para identificar factores determinantes en accidentes de tránsito que afecten a las personas, 2008*.
- [18] A. P. Akgüngör y E. Dogan, *An application of modified Smeed, adapted Andreassen and artificial neural network accident models to three metropolitan cities of Turkey, 2009, pp. 906-913*.
- [19] A. J. Serrano, E. Soria y J. D. Martín, *Redes Neuronales Artificiales, 2009*.
- [20] F. Izaurieta y C. Saavedra, *Redes Neuronales arificiales, Universidad de*

*Concepción, Concepción.*

- [21] *P. Isasi Viñuela y I. Galván León, Redes Neuronales Artificiales Un enfoque Práctico, Pearson Educación, 2004.*
- [22] *C. W. Lira, «Revista Marina En Linea,» [En línea]. Available: <http://www.revistamarina.cl/revistas/1998/5/widow.pdf>.*
- [23] *J. Vergara, Red Sigmoidal Recurrente con Aprendizaje Híbrido para el pronóstico del Volumen de Captura de Anchovetas, 2009.*
- [24] *M. Cohen, H. Franco, N. Morga, D. Rumelhart y V. Abrash, Advances in Neural Information Processing Systems, chapter context-dependent multiple distribution phonetic modeling with MLP's, Morgan Kaufmann, pp. 649-657.*
- [25] *A. Weigend, B. Huberman y D. Rumelhart, Predicting the future: a connectionist approach., PARC, 1990.*
- [26] *W. Yu, H. He y N. Zhang, Advances in Neural Networks-ISNN 2009, Springer, 2009.*
- [27] *M. Mena, Predicción de Capturade Anchoveta Utilizando Redes Wavelets con Optimización de Enjambre de Partículas, 2008.*
- [28] *H. Iba y N. Nikolaev, Adaptive Learning of Polynomial Networks, New York: Springer Science+Business Media, Inc, 2006.*
- [29] *A. Foka, Time Series Prediction Using Evolving Polynomial Neural Networks, 1999.*
- [30] *Y. Karnavas y D. Papadopoulos, EXCITATION CONTROL OF A SYNCHRONOUS MACHINE USING POLYNOMIAL NEURAL NETWORKS, 2004, pp. 169-179.*
- [31] *C. Hervás Martínez, «Nuevos Modelos de Redess Neuronales Evolutivas y Regresión Logística Generalizada utilizando Funciones de Base. Aplicaciones,» 2009.*
- [32] *J. Kennedy y R. Eberhart, «Particle Swarm Optimization,» de In Proceedings of the 1995 IEEE International Conference on Neural Networks, Piscataway, New Jersey, 1995.*
- [33] *J. Q. Lima M. y B. Barán C., «Optimización de Enjambre de Partículas Aplicada al Problem del Cajero Viajante Bi-Objetivo,» Revista Iberoamericana de Inteligencia Artificial, pp. 67-76, 2006.*
- [34] *M. Khanesar, M. Teshnehlab y M. Shoorehdeli, A Novel Binary Particle Swarm Optimization, 2007.*
- [35] *R. Umarani y V. Selvi, Particle Swarm Optimization-Evolution, Overview And Applications. Vol. 12, 2010, pp. 2802-2806.*
- [36] *M. Dorigo, «scholarpedia,» 2008. [En línea]. Available: [http://www.scholarpedia.org/article/Particle\\_swarm\\_optimization](http://www.scholarpedia.org/article/Particle_swarm_optimization).*
- [37] *J. Kennedy y R. Eberhart, A discrete binary version of the particle swarm algorithm, 1997.*
- [38] *M. Settles, An Introduction to Particle Swarm Optimization, University of Idaho, Moscow, 2005.*
- [39] *Clerc, «The swarm and the queen: Towards a deterministic and adaptative particle*

- swarm optimization,» de In Congress on Evolutionary Computation (CEC99), 1999.*
- [40] *M. A. Montes De Oca, J. Peña, T. Stützle, C. Pinciroli y M. Dorigo, IRIDIA – Technical Report Series, 2009.*
- [41] *Y. Chen-Hong, H. Chih-Jen y C. Li-Yeh, «Accelerated Linearly Decreasing Weight Particle Swarm Optimization for Data Clustering, Vol 1,» Hong Kong, 2010.*
- [42] *I. G. Tsoulos y A. Stavrakoudis, Enhancing PSO methods for global optimization, Elsevier, 2010, pp. 1-14.*
- [43] *Y. Liu, Z. Qin, J. Lu y Z. Shi, «Center Particle Swarm Optimization,» de International Conference on Intelligent Computing, Hefei, China, 2005.*
- [44] *Y. Shi y R. Eberhart, «Parameter selection in particle swarm optimization: in Evolutionary Programming VII, Lecture Notes in Computer Science, 1447,» Berlin, 1998.*
- [45] *S. Omkar, R. Khandelwal, T. Ananth, G. Narayana y S. Gopalakrishman, Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures, 2009.*
- [46] *J. Liu, W. Xu y J. Sun, «Quantum-behaved particle swarm optimization with mutation operator,» de Proceedings of 17th international conference on tools with artificial intelligence, Hong Kong, China, 2005.*
- [47] *C. Montt, R. Musso y M. Chacón, Análisis de Accidentes de Tránsito con Métodos de Agrupamiento, 2009.*
- [48] *C. Montt, A. Zúñiga y M. Chacón, «Estructura Bayesianas en Accidentes de Tránsito,» 2009.*

# **Anexos**





## B. Resultados en la elección de nodos ocultos

**Tabla B.1 Resultados MLP con LDWPSO utilizando función tangente hiperbólica para capa oculta y de salida correspondientemente, primeros modelos**

n° test														
n° no	1	2	3	4	5	6	7	8	9	10	Max	Min	Desv.	Prom
1	0,2108	0,2096	0,2117	0,2167	0,2075	0,2129	0,2125	0,2117	0,2117	0,2267	0,2267	0,2075	0,0053	0,2132
2	0,2125	0,2133	0,2117	0,2104	0,2046	0,2029	0,2083	0,2117	0,2067	0,2150	0,2150	0,2029	0,0040	0,2097
3	0,2054	0,2125	0,2038	0,2100	0,2067	0,2063	0,2042	0,2079	0,2058	0,2129	0,2129	0,2038	0,0033	0,2075
4	0,2121	0,2013	0,2013	0,2050	0,2050	0,2008	0,2050	0,2100	0,2088	0,2042	0,2121	0,2008	0,0039	0,2053
5	0,2117	0,2063	0,2108	0,2067	0,2050	0,2233	0,1975	0,2067	0,2158	0,2100	0,2233	0,1975	0,0069	0,2094
6	0,2138	0,2017	0,2200	0,2121	0,2183	0,2175	0,2113	0,2067	0,2071	0,2063	0,2200	0,2017	0,0060	0,2115
7	0,2046	0,2079	0,2096	0,2117	0,2196	0,2050	0,2042	0,1979	0,2054	0,2058	0,2196	0,1979	0,0057	0,2072
8	0,2100	0,2079	0,2358	0,2113	0,2063	0,2133	0,2150	0,2058	0,2017	0,2108	0,2358	0,2017	0,0093	0,2118
9	0,2075	0,2396	0,2354	0,2013	0,2192	0,2092	0,2263	0,2075	0,2396	0,2108	0,2396	0,2013	0,0146	0,2196
10	0,2050	0,2104	0,2038	0,2046	0,2100	0,2058	0,2125	0,2033	0,2196	0,2042	0,2196	0,2033	0,0052	0,2079
11	0,2000	0,2050	0,2063	0,2150	0,2121	0,2183	0,2204	0,2425	0,2146	0,2088	0,2425	0,2000	0,0117	0,2143
12	0,2396	0,2125	0,2096	0,2088	0,2050	0,2296	0,2175	0,2446	0,2150	0,2154	0,2446	0,2050	0,0135	0,2198
13	0,2088	0,2025	0,2046	0,2450	0,2229	0,2163	0,2100	0,2375	0,2129	0,2163	0,2450	0,2025	0,0139	0,2177
14	0,2096	0,2092	0,2208	0,2071	0,2417	0,2029	0,2063	0,2075	0,2154	0,2008	0,2417	0,2008	0,0119	0,2121
15	0,2421	0,2075	0,2458	0,2258	0,2167	0,2017	0,2063	0,2092	0,2142	0,2042	0,2458	0,2017	0,0157	0,2173
16	0,2296	0,2004	0,2417	0,2104	0,2188	0,2221	0,2404	0,2238	0,2200	0,2071	0,2417	0,2004	0,0134	0,2214
17	0,2058	0,2154	0,2075	0,2321	0,2183	0,2008	0,2067	0,2163	0,2454	0,2113	0,2454	0,2008	0,0135	0,2160
18	0,2258	0,2208	0,2000	0,2179	0,2279	0,2096	0,2083	0,2246	0,2179	0,2033	0,2279	0,2000	0,0098	0,2156
19	0,2117	0,2267	0,2517	0,2121	0,2246	0,2183	0,2363	0,2075	0,2058	0,2063	0,2517	0,2058	0,0150	0,2201
20	0,2092	0,2400	0,2517	0,2117	0,2463	0,2525	0,2167	0,2396	0,2321	0,2179	0,2525	0,2092	0,0167	0,2318
21	0,2142	0,2050	0,2654	0,2588	0,2338	0,2300	0,2325	0,2167	0,2396	0,1992	0,2654	0,1992	0,0216	0,2295
22	0,2642	0,2217	0,2000	0,2004	0,2146	0,2079	0,2338	0,2138	0,2138	0,2033	0,2642	0,2000	0,0194	0,2173
23	0,2288	0,2071	0,2083	0,2117	0,2021	0,2104	0,2225	0,2196	0,2375	0,2050	0,2375	0,2021	0,0114	0,2153
24	0,2458	0,2075	0,2229	0,2550	0,2267	0,2117	0,2117	0,2021	0,2275	0,2221	0,2550	0,2021	0,0167	0,2233
25	0,2158	0,2129	0,2071	0,2238	0,2458	0,2138	0,2025	0,2050	0,2054	0,2150	0,2458	0,2025	0,0127	0,2147

**Tabla B.2 Resultados MLP con LDWPSO utilizando función sigmoial y tangente hiperbólica para capa oculta y de salida correspondientemente, primeros modelos**

n° test n° no	1	2	3	4	5	6	7	8	9	10	Max	Min	Desv.	Prom
1	0,4950	0,4950	0,4950	0,4950	0,4950	0,4950	0,4950	0,4950	0,4950	0,4950	0,4950	0,4950	0,0000	0,4950
2	0,1917	0,1967	0,1938	0,1979	0,1967	0,1925	0,1971	0,1992	0,1896	0,1921	0,1992	0,1896	0,0032	0,1947
3	0,1888	0,1938	0,2383	0,1925	0,1892	0,2075	0,1858	0,1854	0,1929	0,1979	0,2383	0,1854	0,0158	0,1972
4	0,2025	0,1967	0,1942	0,2000	0,1842	0,1983	0,1933	0,1883	0,1963	0,1917	0,2025	0,1842	0,0055	0,1945
5	0,1967	0,1971	0,1942	0,1954	0,1917	0,1946	0,1979	0,2021	0,1954	0,2067	0,2067	0,1917	0,0043	0,1972
6	0,2188	0,2196	0,1975	0,1967	0,2217	0,2846	0,1938	0,1938	0,1933	0,1888	0,2846	0,1888	0,0288	0,2108
7	0,1979	0,1908	0,2296	0,1892	0,1979	0,1946	0,2213	0,2258	0,2183	0,2017	0,2296	0,1892	0,0154	0,2067
8	0,1958	0,2200	0,1904	0,1879	0,1979	0,1883	0,2146	0,1892	0,1925	0,1963	0,2200	0,1879	0,0112	0,1973
9	0,2079	0,1988	0,1996	0,2013	0,1958	0,2379	0,2013	0,1892	0,2021	0,1888	0,2379	0,1888	0,0138	0,2023
10	0,2158	0,1871	0,1875	0,2133	0,2046	0,2042	0,2104	0,1896	0,2025	0,1975	0,2158	0,1871	0,0106	0,2013
11	0,2450	0,2021	0,2267	0,1929	0,1863	0,2196	0,1913	0,2383	0,2046	0,2058	0,2450	0,1863	0,0203	0,2113
12	0,1988	0,2204	0,1967	0,1938	0,2150	0,2150	0,2071	0,2038	0,2325	0,1996	0,2325	0,1938	0,0123	0,2083
13	0,1925	0,2158	0,2167	0,2096	0,1983	0,1958	0,2042	0,1904	0,2338	0,2071	0,2338	0,1904	0,0133	0,2064
14	0,2008	0,2046	0,2283	0,2000	0,1942	0,1992	0,2108	0,1996	0,1896	0,2196	0,2283	0,1896	0,0118	0,2047
15	0,1992	0,1992	0,2071	0,1921	0,1867	0,2242	0,2304	0,2004	0,1950	0,1904	0,2304	0,1867	0,0144	0,2025
16	0,1950	0,2300	0,2025	0,2200	0,2013	0,1946	0,2029	0,1967	0,2492	0,2054	0,2492	0,1946	0,0179	0,2098
17	0,2200	0,2008	0,2442	0,2408	0,1908	0,2025	0,1933	0,1971	0,2171	0,2188	0,2442	0,1908	0,0190	0,2125
18	0,2342	0,2154	0,1946	0,2125	0,1975	0,2058	0,1975	0,1967	0,2346	0,1921	0,2346	0,1921	0,0158	0,2081
19	0,1933	0,2142	0,2438	0,2042	0,2108	0,1946	0,2021	0,2329	0,2538	0,1996	0,2538	0,1933	0,0213	0,2149
20	0,1933	0,2275	0,2179	0,2121	0,2104	0,2563	0,1975	0,2213	0,2000	0,1946	0,2563	0,1933	0,0192	0,2131
21	0,2246	0,2013	0,1921	0,2038	0,1917	0,2267	0,2346	0,2079	0,2042	0,1963	0,2346	0,1917	0,0152	0,2083
22	0,1879	0,2296	0,2054	0,1896	0,1929	0,2108	0,1813	0,2296	0,2042	0,2129	0,2296	0,1813	0,0168	0,2044
23	0,1892	0,2063	0,2125	0,2346	0,1996	0,2321	0,2042	0,2529	0,2108	0,1975	0,2529	0,1892	0,0198	0,2140
24	0,2396	0,2000	0,2138	0,2263	0,1992	0,1917	0,2096	0,1933	0,1996	0,2029	0,2396	0,1917	0,0152	0,2076
25	0,2204	0,1954	0,1967	0,2017	0,1988	0,2250	0,1971	0,1958	0,2129	0,2433	0,2433	0,1954	0,0163	0,2087

**Tabla B.3 Resultados MLP con LDWPSO utilizando función sigmoial para capa oculta y de salida, primeros modelos**

n° test n° no	1	2	3	4	5	6	7	8	9	10	Max	Min	Desv.	Prom
1	0,2192	0,1950	0,2192	0,2342	0,2150	0,2046	0,2217	0,2038	0,2346	0,2342	0,2346	0,1950	0,0139	0,2181
2	0,2033	0,2317	0,2258	0,2021	0,1854	0,2050	0,2154	0,2350	0,2104	0,2279	0,2350	0,1854	0,0158	0,2142
3	0,1992	0,1958	0,2042	0,1963	0,1963	0,2142	0,1996	0,2217	0,2004	0,1917	0,2217	0,1917	0,0092	0,2019
4	0,2021	0,2121	0,1992	0,2046	0,1979	0,1946	0,1971	0,2025	0,2046	0,2042	0,2121	0,1946	0,0050	0,2019
5	0,1958	0,1963	0,2050	0,2000	0,2038	0,2046	0,2079	0,2083	0,1983	0,1925	0,2083	0,1925	0,0054	0,2013
6	0,2004	0,2117	0,2017	0,2042	0,2158	0,2071	0,2133	0,2154	0,2058	0,1958	0,2158	0,1958	0,0068	0,2071
7	0,2013	0,2146	0,1942	0,2075	0,2208	0,1992	0,1975	0,2213	0,1867	0,2038	0,2213	0,1867	0,0114	0,2047
8	0,2275	0,2138	0,1983	0,2100	0,2142	0,2004	0,1975	0,2113	0,2054	0,2029	0,2275	0,1975	0,0092	0,2081
9	0,2246	0,1971	0,2267	0,2096	0,2021	0,2183	0,2096	0,2029	0,1971	0,2017	0,2267	0,1971	0,0109	0,2090
10	0,2088	0,2108	0,2113	0,2400	0,2408	0,2313	0,2021	0,2408	0,2054	0,2188	0,2408	0,2021	0,0157	0,2210
11	0,2388	0,2146	0,1963	0,2071	0,2083	0,2042	0,2067	0,2175	0,1958	0,2029	0,2388	0,1958	0,0124	0,2092
12	0,2029	0,2321	0,2283	0,1942	0,2225	0,2071	0,1983	0,2279	0,2208	0,2221	0,2321	0,1942	0,0137	0,2156
13	0,2050	0,2208	0,2238	0,2108	0,1963	0,2854	0,2004	0,2013	0,2042	0,2104	0,2854	0,1963	0,0260	0,2158
14	0,2163	0,1925	0,2258	0,1933	0,2096	0,2129	0,2125	0,2217	0,2129	0,2188	0,2258	0,1925	0,0110	0,2116
15	0,2325	0,2054	0,1958	0,2296	0,2004	0,1967	0,2071	0,2271	0,2271	0,2167	0,2325	0,1958	0,0144	0,2138
16	0,2492	0,2054	0,2346	0,2208	0,2104	0,2271	0,2083	0,1946	0,1871	0,2104	0,2492	0,1871	0,0186	0,2148
17	0,2108	0,2254	0,2304	0,2029	0,2042	0,2175	0,2329	0,2071	0,2450	0,2304	0,2450	0,2029	0,0143	0,2207
18	0,2121	0,1996	0,2325	0,2179	0,1946	0,2263	0,2167	0,1975	0,2221	0,2513	0,2513	0,1946	0,0174	0,2170
19	0,2146	0,2254	0,1967	0,2067	0,2038	0,2375	0,1996	0,2042	0,2067	0,2463	0,2463	0,1967	0,0168	0,2141
20	0,2050	0,2179	0,2113	0,2096	0,2421	0,2063	0,2421	0,2404	0,2104	0,2000	0,2421	0,2000	0,0165	0,2185
21	0,2358	0,2008	0,2038	0,1917	0,2013	0,2054	0,2058	0,2138	0,1988	0,2033	0,2358	0,1917	0,0119	0,2060
22	0,1996	0,2317	0,2204	0,2150	0,2142	0,2171	0,2179	0,2379	0,2233	0,2450	0,2450	0,1996	0,0131	0,2222
23	0,2146	0,1925	0,2213	0,2421	0,2063	0,2367	0,2204	0,2046	0,2033	0,1996	0,2421	0,1925	0,0161	0,2141
24	0,2013	0,2038	0,2013	0,2179	0,2267	0,2225	0,2129	0,2075	0,2304	0,2525	0,2525	0,2013	0,0161	0,2177
25	0,2267	0,2138	0,2050	0,2008	0,2254	0,2625	0,2088	0,2038	0,2175	0,2050	0,2625	0,2008	0,0184	0,2169

**Tabla B.4 Resultados MLP con LDWPSO utilizando función tangente hiperbólica y sigmoïdal para capa oculta y de salida correspondientemente, primeros modelos**

n° test n° no	1	2	3	4	5	6	7	8	9	10	Max	Min	Desv.	Prom
1	0,2079	0,2079	0,1983	0,2071	0,2058	0,2050	0,2025	0,2100	0,2075	0,2138	0,2138	0,1983	0,0042	0,2066
2	0,2117	0,2063	0,2004	0,1988	0,2183	0,2038	0,2029	0,2054	0,2067	0,2217	0,2217	0,1988	0,0075	0,2076
3	0,2096	0,1971	0,2121	0,2092	0,2029	0,2021	0,2038	0,1983	0,2271	0,2346	0,2346	0,1971	0,0123	0,2097
4	0,2071	0,1975	0,1992	0,2025	0,2029	0,2008	0,2021	0,2004	0,2171	0,2117	0,2171	0,1975	0,0061	0,2041
5	0,2267	0,2004	0,1996	0,2071	0,2096	0,1988	0,2175	0,2000	0,2221	0,2042	0,2267	0,1988	0,0102	0,2086
6	0,2063	0,2125	0,2654	0,1967	0,2000	0,2067	0,2083	0,2104	0,2329	0,2083	0,2654	0,1967	0,0202	0,2148
7	0,2083	0,2063	0,2558	0,2108	0,2025	0,2029	0,2688	0,2017	0,2092	0,2325	0,2688	0,2017	0,0242	0,2199
8	0,2046	0,2092	0,2050	0,2083	0,2150	0,2083	0,2433	0,1992	0,1996	0,2021	0,2433	0,1992	0,0128	0,2095
9	0,2071	0,2029	0,2100	0,2125	0,2088	0,2054	0,2138	0,2042	0,2083	0,2300	0,2300	0,2029	0,0077	0,2103
10	0,2004	0,2383	0,2129	0,2488	0,2183	0,2150	0,2013	0,2017	0,2121	0,2058	0,2488	0,2004	0,0162	0,2155
11	0,2275	0,2146	0,2250	0,2046	0,2046	0,2246	0,2417	0,2021	0,2388	0,2533	0,2533	0,2021	0,0174	0,2237
12	0,2083	0,2029	0,2033	0,2221	0,2033	0,2058	0,2167	0,2113	0,2033	0,2271	0,2271	0,2029	0,0087	0,2104
13	0,2042	0,2125	0,2054	0,2233	0,1971	0,1992	0,2283	0,2142	0,2046	0,2071	0,2283	0,1971	0,0101	0,2096
14	0,2150	0,2121	0,2371	0,2033	0,2054	0,2100	0,2021	0,2071	0,2017	0,2129	0,2371	0,2017	0,0104	0,2107
15	0,2333	0,2442	0,2075	0,2017	0,2288	0,2067	0,2213	0,2079	0,2113	0,2138	0,2442	0,2017	0,0138	0,2176
16	0,2238	0,2254	0,2075	0,2133	0,2121	0,2050	0,2125	0,2329	0,2154	0,2346	0,2346	0,2050	0,0103	0,2183
17	0,2204	0,2138	0,2129	0,2117	0,2179	0,2083	0,2133	0,2129	0,2067	0,2367	0,2367	0,2067	0,0085	0,2155
18	0,2067	0,1979	0,2142	0,2313	0,2071	0,2088	0,2225	0,2217	0,2163	0,2142	0,2313	0,1979	0,0096	0,2140
19	0,2596	0,2088	0,2221	0,2463	0,2008	0,2104	0,2171	0,2292	0,2079	0,2204	0,2596	0,2008	0,0184	0,2223
20	0,2392	0,2063	0,2104	0,2329	0,2117	0,2071	0,2146	0,2279	0,2175	0,2371	0,2392	0,2063	0,0126	0,2205
21	0,2263	0,2238	0,2392	0,2079	0,2213	0,2258	0,2333	0,2188	0,2304	0,1954	0,2392	0,1954	0,0127	0,2222
22	0,2104	0,2092	0,2042	0,2225	0,2096	0,2046	0,2183	0,2458	0,2038	0,2038	0,2458	0,2038	0,0131	0,2132
23	0,2050	0,2263	0,2100	0,2079	0,2058	0,2213	0,2154	0,2146	0,2404	0,2179	0,2404	0,2050	0,0108	0,2165
24	0,2208	0,2050	0,1967	0,2125	0,2096	0,2229	0,2033	0,2154	0,2196	0,2029	0,2229	0,1967	0,0088	0,2109
25	0,2204	0,2025	0,2333	0,2146	0,2113	0,2242	0,2171	0,2313	0,2213	0,2108	0,2333	0,2025	0,0095	0,2187

**Tabla B.5 Resultados de algoritmos PSO para elección de nodos ocultos, utilizando función de activación tangente hiperbólica y sigmoïdal para capa oculta y de salida correspondientemente, segundos modelos propuestos**

n° no	LDWPSO				QPSO				LDWQPS			
	Max	Min	Desv.	Prom	Max	Min	Desv.	Prom	Max	Min	Desv.	Prom
1	0,4533	0,4424	0,0033	0,4464	0,5446	0,4669	0,0252	0,5071	0,6107	0,4967	0,0320	0,5302
2	0,4445	0,4291	0,0046	0,4331	0,5629	0,4423	0,0349	0,4651	0,5098	0,4591	0,0148	0,4716
3	0,4430	0,4252	0,0060	0,4346	0,5014	0,4508	0,0152	0,4608	0,5467	0,4583	0,0268	0,4749
4	0,4420	0,4306	0,0037	0,4356	0,5395	0,4453	0,0310	0,4667	0,5515	0,4629	0,0319	0,4953
5	0,4392	0,4241	0,0056	0,4315	0,6057	0,4381	0,0490	0,4672	0,5525	0,4547	0,0289	0,4776
6	0,4959	0,4287	0,0202	0,4409	0,5409	0,4382	0,0289	0,4603	0,5130	0,4559	0,0213	0,4718
7	0,4421	0,4265	0,0051	0,4345	0,5697	0,4413	0,0383	0,4680	0,5319	0,4461	0,0251	0,4816
8	0,4483	0,4266	0,0075	0,4341	0,5179	0,4382	0,0250	0,4578	0,6011	0,4592	0,0426	0,4872
9	0,4409	0,4296	0,0032	0,4362	0,4780	0,4465	0,0098	0,4577	0,5346	0,4592	0,0226	0,4873
10	0,4381	0,4261	0,0032	0,4321	0,5945	0,4470	0,0524	0,4781	0,5499	0,4494	0,0341	0,4815
11	0,4496	0,4261	0,0073	0,4358	0,5530	0,4354	0,0398	0,4839	0,5510	0,4515	0,0351	0,4872
12	0,4427	0,4240	0,0062	0,4337	0,5762	0,4449	0,0485	0,4890	0,5422	0,4589	0,0245	0,4886

**Tabla B. 6 Resultados de algoritmos PSO para elección de nodos ocultos, utilizando función de activación sigmoïdal para capa oculta y de salida, segundos modelos propuestos**

n° no	WPSO				QPSO				WQPS			
	Max	Min	Desv.	Prom	Max	Min	Desv.	Prom	Max	Min	Desv.	Prom
1	0,5128	0,5097	0,0010	0,5103	0,5256	0,5235	0,0006	0,5245	0,6114	0,4956	0,0359	0,5093
2	0,5101	0,4308	0,0361	0,4671	0,5313	0,4327	0,0396	0,4680	0,5372	0,4162	0,0440	0,4610
3	0,4480	0,4299	0,0063	0,4396	0,4708	0,4296	0,0130	0,4444	0,5060	0,4185	0,0355	0,4554
4	0,5045	0,4283	0,0217	0,4445	0,5067	0,4349	0,0226	0,4534	0,5054	0,4168	0,0347	0,4506
5	0,5081	0,4303	0,0234	0,4421	0,4656	0,4237	0,0119	0,4432	0,5773	0,4242	0,0474	0,4549
6	0,4447	0,4278	0,0056	0,4388	0,6083	0,4309	0,0520	0,4713	0,6018	0,4296	0,0582	0,4742
7	0,4560	0,4308	0,0072	0,4400	0,5172	0,4273	0,0259	0,4482	0,5176	0,4263	0,0302	0,4546
8	0,4477	0,4265	0,0058	0,4402	0,4741	0,4308	0,0135	0,4470	0,5586	0,4241	0,0443	0,4580
9	0,4488	0,4290	0,0058	0,4351	0,5907	0,4326	0,0612	0,4707	0,5729	0,4196	0,0548	0,4703
10	0,4585	0,4288	0,0086	0,4395	0,6400	0,4332	0,0809	0,5309	0,6309	0,4191	0,0642	0,4926
11	0,4499	0,4274	0,0073	0,4380	0,5923	0,4405	0,0627	0,4962	0,6189	0,4257	0,0783	0,5209
12	0,4496	0,4321	0,0050	0,4394	0,6052	0,4344	0,0695	0,5232	0,6274	0,4161	0,0715	0,4936

**Tabla B.7 Resultados de algoritmos PSO para elección de nodos ocultos, utilizando función de activación sigmoïdal para la capa oculta y softmax para la capa de salida, terceros modelos**

n° no	LDWPSO				QPSO				LDWQPSO			
	Max	Min	Desv.	Prom	Max	Min	Desv.	Prom	Max	Min	Desv.	Prom
1	0,6526	0,6485	0,0012	0,6492	0,6559	0,5843	0,0225	0,6146	0,6415	0,5753	0,0190	0,6035
2	0,6476	0,5955	0,0209	0,6249	0,6384	0,5251	0,0343	0,5973	0,6225	0,5150	0,0404	0,5823
3	0,6491	0,5772	0,0268	0,6109	0,6123	0,5213	0,0306	0,5576	0,5990	0,5108	0,0319	0,5564
4	0,6272	0,5687	0,0196	0,5893	0,6236	0,5214	0,0348	0,5806	0,6290	0,5278	0,0268	0,5862
5	0,6213	0,5686	0,0169	0,5897	0,6102	0,5467	0,0217	0,5850	0,6640	0,5155	0,0543	0,5857
6	0,6008	0,5623	0,0129	0,5804	0,6103	0,5160	0,0362	0,5685	0,6250	0,5185	0,0367	0,5783
7	0,6035	0,5589	0,0151	0,5823	0,6418	0,5501	0,0261	0,5855	0,6386	0,5115	0,0463	0,5611
8	0,6004	0,5539	0,0166	0,5782	0,6325	0,5144	0,0344	0,5695	0,6451	0,5151	0,0392	0,5745
9	0,5982	0,5601	0,0102	0,5808	0,6212	0,5141	0,0394	0,5672	0,6052	0,5119	0,0354	0,5684
10	0,5979	0,5617	0,0135	0,5767	0,6191	0,5426	0,0273	0,5801	0,6099	0,5129	0,0271	0,5765
11	0,5951	0,5396	0,0154	0,5670	0,6280	0,5310	0,0324	0,5831	0,6098	0,5155	0,0299	0,5684
12	0,5896	0,5479	0,0147	0,5621	0,6217	0,5598	0,0203	0,5924	0,6126	0,5103	0,0361	0,5698
13	0,5912	0,5509	0,0118	0,5656	0,6271	0,5169	0,0310	0,5739	0,6103	0,5140	0,0355	0,5747
14	0,5981	0,5306	0,0213	0,5666	0,6201	0,5220	0,0321	0,5772	0,6219	0,5058	0,0405	0,5731
15	0,5867	0,5474	0,0138	0,5667	0,6390	0,5198	0,0417	0,5798	0,6087	0,5502	0,0199	0,5792
16	0,5842	0,5341	0,0169	0,5613	0,6260	0,5224	0,0364	0,5696	0,6143	0,5108	0,0388	0,5657
17	0,5861	0,5338	0,0168	0,5607	0,6304	0,5678	0,0221	0,5914	0,6027	0,5120	0,0313	0,5602
18	0,5830	0,5380	0,0144	0,5572	0,6493	0,5153	0,0432	0,5812	0,6222	0,5109	0,0336	0,5681
19	0,5830	0,5320	0,0173	0,5523	0,5876	0,5138	0,0237	0,5649	0,6232	0,5393	0,0250	0,5745
20	0,5693	0,5233	0,0155	0,5502	0,6239	0,5196	0,0377	0,5850	0,6057	0,5112	0,0334	0,5572
21	0,5676	0,5395	0,0096	0,5551	0,6190	0,5074	0,0327	0,5768	0,6267	0,5233	0,0348	0,5752
22	0,5630	0,5246	0,0116	0,5465	0,6256	0,5117	0,0380	0,5555	0,6170	0,5336	0,0254	0,5748
23	0,5952	0,5294	0,0187	0,5506	0,6284	0,5170	0,0332	0,5731	0,6107	0,5333	0,0212	0,5684
24	0,5938	0,5376	0,0171	0,5470	0,6103	0,5468	0,0224	0,5787	0,6068	0,5126	0,0292	0,5671
25	0,6271	0,5235	0,0309	0,5482	0,6136	0,5199	0,0295	0,5610	0,6140	0,5227	0,0256	0,5775
26	0,5631	0,5192	0,0161	0,5421	0,6015	0,5116	0,0305	0,5594	0,6178	0,5284	0,0236	0,5772
27	0,5702	0,5193	0,0152	0,5436	0,5982	0,5141	0,0309	0,5637	0,5977	0,5096	0,0329	0,5523
28	0,5676	0,5174	0,0139	0,5418	0,6074	0,5151	0,0305	0,5745	0,6081	0,5252	0,0263	0,5671
29	0,5576	0,5240	0,0100	0,5354	0,6122	0,5187	0,0314	0,5597	0,5869	0,5050	0,0240	0,5579
30	0,5625	0,5139	0,0157	0,5403	0,6216	0,5193	0,0338	0,5722	0,6143	0,5061	0,0325	0,5676
31	0,5630	0,5302	0,0105	0,5451	0,6225	0,5373	0,0266	0,5788	0,6193	0,5139	0,0324	0,5700
32	0,5479	0,5166	0,0102	0,5316	0,6265	0,5269	0,0300	0,5748	0,6007	0,5183	0,0266	0,5696
33	0,5503	0,5190	0,0092	0,5347	0,5924	0,5072	0,0260	0,5553	0,6104	0,5395	0,0233	0,5716
34	0,5610	0,5216	0,0117	0,5392	0,6100	0,5100	0,0289	0,5693	0,6072	0,5133	0,0285	0,5670
35	0,5462	0,5069	0,0112	0,5342	0,6300	0,5179	0,0357	0,5687	0,5970	0,5200	0,0244	0,5717
36	0,5973	0,5202	0,0234	0,5385	0,6062	0,5124	0,0328	0,5707	0,6195	0,5142	0,0308	0,5776
37	0,5540	0,5066	0,0148	0,5305	0,6078	0,5148	0,0289	0,5602	0,6227	0,5292	0,0306	0,5720
38	0,5512	0,5150	0,0099	0,5285	0,5902	0,5148	0,0282	0,5619	0,6124	0,5348	0,0203	0,5741
39	0,5393	0,5157	0,0092	0,5287	0,5863	0,5443	0,0155	0,5658	0,5987	0,5181	0,0284	0,5689

40	0,5507	0,5123	0,0129	0,5288	0,6139	0,5207	0,0238	0,5717	0,6056	0,5337	0,0254	0,5697
41	0,5333	0,5061	0,0089	0,5236	0,6128	0,5199	0,0330	0,5714	0,6044	0,5479	0,0165	0,5688
42	0,5495	0,5107	0,0130	0,5293	0,6409	0,5132	0,0400	0,5564	0,6110	0,5164	0,0291	0,5737
43	0,5412	0,5087	0,0106	0,5259	0,6112	0,5460	0,0221	0,5770	0,6185	0,5403	0,0201	0,5736
44	0,5423	0,5072	0,0114	0,5227	0,6053	0,5219	0,0257	0,5617	0,6016	0,5169	0,0295	0,5599
45	0,5371	0,5048	0,0120	0,5236	0,6030	0,5199	0,0237	0,5710	0,5917	0,5254	0,0209	0,5738
46	0,5528	0,5083	0,0123	0,5294	0,5971	0,5401	0,0168	0,5651	0,5859	0,5114	0,0267	0,5568
47	0,5543	0,5070	0,0141	0,5242	0,6094	0,5164	0,0301	0,5668	0,5877	0,5205	0,0194	0,5670
48	0,5451	0,5141	0,0106	0,5256	0,6057	0,5189	0,0283	0,5622	0,5915	0,5410	0,0189	0,5646
49	0,5614	0,5037	0,0172	0,5266	0,5989	0,5145	0,0244	0,5563	0,6220	0,5211	0,0286	0,5684
50	0,5321	0,5128	0,0067	0,5242	0,5794	0,5081	0,0261	0,5493	0,5910	0,5121	0,0221	0,5560

## C. Resultados en el entrenamiento de la red

**Tabla C.1 Resultados del entrenamiento para MLP con LWPSO con 4 nodos ocultos y función tangente hiperbólica para capa oculta y de salida, primeros modelos**

n° test	Error	Exactitud	Especificidad	Sensibilidad
1	0,1917	0,8083	0,8690	0,7486
2	0,1921	0,8079	0,9543	0,6762
3	0,1879	0,8121	0,9461	0,6851
4	0,2000	0,8000	0,9554	0,6414
5	0,1871	0,8129	0,9109	0,7165
6	0,2325	0,7675	0,7470	0,7865
7	0,1808	0,8192	0,9513	0,6887
8	0,1929	0,8071	0,9434	0,6743
9	0,1896	0,8104	0,9510	0,6689
10	0,1879	0,8121	0,9492	0,6834
Max	0,2325	0,8192	0,9554	0,7865
Min	0,1808	0,7675	0,7470	0,6414
Desv.	0,0143	0,0143	0,0659	0,0425
Prom.	0,1943	0,8058	0,9178	0,6970

**Tabla C.2 Resultados del entrenamiento para MLP con LWPSO con 4 nodos ocultos y función de sigmooidal y tangente hiperbólica para capa oculta y de salida correspondientemente, primeros modelos**

n° test	Error	Exactitud	Especificidad	Sensibilidad
1	0,1875	0,8125	0,8974	0,7381
2	0,1950	0,8050	0,9399	0,6811
3	0,1817	0,8183	0,9480	0,6904
4	0,1942	0,8058	0,9333	0,6907
5	0,1863	0,8138	0,9660	0,6672
6	0,1875	0,8125	0,9682	0,6661
7	0,1888	0,8113	0,9397	0,6877
8	0,2008	0,7992	0,9340	0,6686
9	0,1742	0,8258	0,9439	0,7054
10	0,1921	0,8079	0,9498	0,6716
Max	0,2008	0,8258	0,9682	0,7381
Min	0,1742	0,7992	0,8974	0,6661
Desv.	0,0074	0,0074	0,0197	0,0221
Prom.	0,1888	0,8112	0,9420	0,6867

**Tabla C.3 Resultados del entrenamiento para MLP con LWDPPO con 5 nodos ocultos con función de activación sigmoïdal para capa oculta y de salida, primeros modelos**

n° test	Error	Exactitud	Especificidad	Sensibilidad
1	0,1883	0,8117	0,8784	0,7459
2	0,2008	0,7992	0,9107	0,6793
3	0,1808	0,8192	0,9407	0,7047
4	0,1838	0,8163	0,8775	0,7566
5	0,2050	0,7950	0,8804	0,7101
6	0,1883	0,8117	0,9431	0,6849
7	0,1892	0,8108	0,9254	0,6976
8	0,1858	0,8142	0,9329	0,6970
9	0,1942	0,8058	0,9599	0,6591
10	0,1979	0,8021	0,9480	0,6629
Max	0,2050	0,8192	0,9599	0,7566
Min	0,1808	0,7950	0,8775	0,6591
Desv.	0,0078	0,0078	0,0311	0,0319
Prom.	0,1914	0,8086	0,9197	0,6998

**Tabla C.4 Resultados del entrenamiento para MLP con LWDPPO con 4 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, primeros modelos**

n° test	Error	Exactitud	Especificidad	Sensibilidad
1	0,1820	0,8179	0,9619	0,6778
2	0,1825	0,8175	0,9591	0,6822
3	0,1904	0,8096	0,9464	0,6744
4	0,1838	0,8163	0,9639	0,6711
5	0,1971	0,8029	0,9050	0,6989
6	0,1979	0,8021	0,9530	0,6588
7	0,1875	0,8125	0,9631	0,6574
8	0,2050	0,7950	0,9598	0,6252
9	0,1846	0,8154	0,9384	0,6989
10	0,1888	0,8113	0,9263	0,7024
Max	0,2050	0,8179	0,9639	0,7024
Min	0,1821	0,7950	0,9050	0,6252
Desv.	0,0077	0,0077	0,0194	0,0236
Prom.	0,1900	0,8100	0,9477	0,6747

**Tabla C.5 Resultados del entrenamiento para MLP con LDWPSO con 5 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	1000	5	0,4088	0,8161	0,9407	0,7061
LDWPSO	3000	5	0,4476	0,8008	0,9441	0,6637
LDWPSO	10000	5	0,4420	0,8035	0,9315	0,6812
LDWPSO	20000	5	0,4409	0,8036	0,9502	0,6628
Promedio			0,4348	0,8060	0,9416	0,6785

**Tabla C.6 Resultados del entrenamiento para MLP con QPSO con 9 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
QPSO	1000	9	0,4859	0,7825	0,9287	0,6342
QPSO	3000	9	0,4536	0,8025	0,9636	0,6396
QPSO	10000	9	0,4564	0,7926	0,9242	0,6657
QPSO	20000	9	0,4503	0,7981	0,9467	0,6564
Promedio			0,4616	0,7939	0,9408	0,6490

**Tabla C.7 Resultados del entrenamiento para MLP con QPSO con 9 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWQPSO	1000	2	0,4862	0,7600	0,8978	0,6432
LDWQPSO	3000	2	0,4734	0,7804	0,9323	0,6392
LDWQPSO	10000	2	0,4654	0,7886	0,9653	0,6234
LDWQPSO	20000	2	0,4533	0,7887	0,9731	0,6097
Promedio			0,4696	0,7794	0,9421	0,6289

**Tabla C.8 Resultados del entrenamiento para MLP con LDWPSO con 9 nodos ocultos con función sigmoïdal para la capa oculta y de salida, utilizando validaci3n cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	1000	9	0,4569	0,7888	0,9152	0,6749
LDWPSO	3000	9	0,4506	0,8000	0,9447	0,6512
LDWPSO	10000	9	0,4303	0,8076	0,9296	0,6902
LDWPSO	20000	9	0,4372	0,8058	0,9416	0,6756
Promedio			0,4438	0,8006	0,9328	0,6730

**Tabla C.9 Resultados del entrenamiento para MLP con QPSO con 5 nodos ocultos con funci3n sigmoïdal para la capa oculta y de salida, utilizando validaci3n cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
QPSO	1000	5	0,4814	0,7475	0,8209	0,6731
QPSO	3000	5	0,4175	0,8254	0,9637	0,6845
QPSO	10000	5	0,4616	0,7930	0,9249	0,6636
QPSO	20000	5	0,4448	0,8004	0,9513	0,6525
Promedio			0,4513	0,7916	0,9152	0,6684

**Tabla C.10 Resultados del entrenamiento para MLP con LDWQPSO con 4 nodos ocultos con funci3n sigmoïdal para la capa oculta y de salida, utilizando validaci3n cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWQPSO	1000	4	0,4697	0,7588	0,8560	0,6824
LDWQPSO	3000	4	0,4765	0,7871	0,9306	0,6377
LDWQPSO	10000	4	0,4530	0,7965	0,9556	0,6446
LDWQPSO	20000	4	0,4606	0,7782	0,9054	0,6565
Promedio			0,4650	0,7802	0,9119	0,6553

**Tabla C.11 Resultados del entrenamiento para MLP con LDWPSO con 44 nodos ocultos con función sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	1000	44	0,4658	0,8475	0,9020	0,7864
LDWPSO	3000	44	0,4845	0,8288	0,9040	0,7511
LDWPSO	10000	44	0,5164	0,7969	0,8732	0,7219
LDWPSO	20000	44	0,5184	0,7949	0,8740	0,7171
Promedio			0,4963	0,8170	0,8883	0,7441

**Tabla C.12 Resultados del entrenamiento para MLP con QPSO con 50 nodos ocultos con función sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
QPSO	1000	50	0,5145	0,7988	0,8669	0,7184
QPSO	3000	50	0,5491	0,7642	0,8206	0,7044
QPSO	10000	50	0,5590	0,7543	0,8488	0,6612
QPSO	20000	50	0,5643	0,7489	0,8483	0,6524
Promedio			0,5467	0,7666	0,8462	0,6841

**Tabla C.13 Resultados del entrenamiento para MLP con LWDQPSO con 27 nodos ocultos con función sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LWDQPSO	1000	27	0,5033	0,8100	0,8420	0,7671
LWDQPSO	3000	27	0,5412	0,7721	0,8442	0,7018
LWDQPSO	10000	27	0,5598	0,7535	0,8477	0,6572
LWDQPSO	20000	27	0,5625	0,7508	0,8396	0,6643
Promedio			0,5417	0,7716	0,8434	0,6976

## D. Resultados en la evaluación de la red

**Tabla D.1 Resultados de la evaluación para MLP con LWDPPO con 4 nodos ocultos y función tangente hiperbólica para capa oculta y de salida, primeros modelos**

n° test	Error	Exactitud	Especificidad	Sensibilidad
1	0,2133	0,7867	0,8425	0,7401
2	0,1850	0,8150	0,9589	0,6549
3	0,1917	0,8083	0,9288	0,7022
4	0,1783	0,8217	0,9545	0,7006
5	0,1833	0,8167	0,8889	0,7459
6	0,2267	0,7733	0,7492	0,7986
7	0,1767	0,8233	0,9642	0,6758
8	0,1883	0,8117	0,9244	0,7055
9	0,1950	0,8050	0,9529	0,6601
10	0,1950	0,8050	0,9226	0,6553
Max	0,2267	0,8233	0,9642	0,7986
Min	0,1767	0,7733	0,7492	0,6549
Desv.	0,0157	0,0157	0,0673	0,0466
Prom.	0,1933	0,8067	0,9087	0,7039

**Tabla D.2 Resultados de la evaluación para MLP con LWDPPO con 4 nodos ocultos y función de sigmooidal y tangente hiperbólica para capa oculta y de salida correspondientemente, primeros modelos**

n° test	Error	Exactitud	Especificidad	Sensibilidad
1	0,2000	0,8000	0,8872	0,7305
2	0,2150	0,7850	0,9490	0,6275
3	0,2167	0,7833	0,9014	0,6699
4	0,1900	0,8100	0,9412	0,6735
5	0,2033	0,7967	0,9617	0,6454
6	0,1833	0,8167	0,9490	0,6895
7	0,1883	0,8117	0,9270	0,6842
8	0,1767	0,8233	0,9592	0,6928
9	0,1800	0,8200	0,9223	0,7113
10	0,1767	0,8233	0,9730	0,6776
Max	0,2167	0,8233	0,9730	0,7305
Min	0,1767	0,7833	0,8872	0,6275
Desv.	0,0150	0,0150	0,0274	0,0297
Prom.	0,1930	0,8070	0,9371	0,6802

**Tabla D.3 Resultados de la evaluación para MLP con LWDPPO con 5 nodos ocultos con función de activación sigmoïdal para capa oculta y de salida, primeros modelos**

n° test	Error	Exactitud	Especificidad	Sensibilidad
1	0,2267	0,7733	0,8604	0,6815
2	0,2200	0,7800	0,8729	0,6877
3	0,1983	0,8017	0,9536	0,6477
4	0,2267	0,7733	0,8664	0,6758
5	0,2367	0,7633	0,8807	0,6571
6	0,1667	0,8333	0,9609	0,7210
7	0,1883	0,8117	0,9057	0,7195
8	0,1917	0,8083	0,8938	0,7273
9	0,2017	0,7983	0,9452	0,6591
10	0,1983	0,8017	0,9387	0,6552
Max	0,2367	0,8333	0,9609	0,7273
Min	0,1667	0,7633	0,8604	0,6477
Desv.	0,0216	0,0216	0,0386	0,0300
Prom.	0,2055	0,7945	0,9078	0,6832

**Tabla D.4 Resultados de la evaluación para MLP con LWDPPO con 4 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, primeros modelos**

n° test	Error	Exactitud	Especificidad	Sensibilidad
1	0,1967	0,8033	0,9601	0,6455
2	0,1950	0,8050	0,9449	0,6890
3	0,1967	0,8033	0,9699	0,6379
4	0,1750	0,8250	0,9832	0,6689
5	0,2100	0,7900	0,9076	0,6700
6	0,2050	0,7950	0,9345	0,6645
7	0,2167	0,7833	0,9552	0,6446
8	0,2083	0,7917	0,9477	0,6293
9	0,1850	0,8150	0,9263	0,7143
10	0,2100	0,7900	0,9063	0,6827
Max	0,2167	0,8250	0,9832	0,7143
Min	0,1750	0,7833	0,9063	0,6293
Desv.	0,0128	0,0128	0,0253	0,0262
Prom.	0,1998	0,8002	0,9436	0,6647

**Tabla D.5 Resultados de la evaluación para MLP con LDWPSO con 5 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	1000	5	0,4350	0,8150	0,9057	0,7128
LDWPSO	3000	5	0,4296	0,8017	0,9519	0,6602
LDWPSO	10000	5	0,4290	0,8100	0,9476	0,6796
LDWPSO	20000	5	0,4214	0,8070	0,9521	0,6704
Promedio			0,4288	0,8084	0,9393	0,6808

**Tabla D.6 Resultados de la evaluación para MLP con QPSO con 9 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
QPSO	1000	9	0,3454	0,8350	0,8866	0,7864
QPSO	3000	9	0,4611	0,7950	0,9634	0,6544
QPSO	10000	9	0,4387	0,8065	0,9423	0,6693
QPSO	20000	9	0,4412	0,8077	0,9495	0,6712
Promedio			0,4216	0,8111	0,9355	0,6953

**Tabla D.7 Resultados de la evaluación para MLP con LDWQPSO con 9 nodos ocultos con función de activación tangente hiperbólica y sigmoïdal para la capa oculta y de salida correspondientemente, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWQPSO	1000	2	0,4628	0,7950	0,9709	0,6082
LDWQPSO	3000	2	0,4929	0,7717	0,9492	0,6000
LDWQPSO	10000	2	0,4718	0,8135	0,9707	0,6489
LDWQPSO	20000	2	0,4533	0,7887	0,9731	0,6097
Promedio			0,4702	0,7922	0,9660	0,6167

**Tabla D.8 Resultados de la evaluación para MLP con LDWPSO con 9 nodos ocultos con función de activación sigmoïdal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	1000	9	0,5145	0,7300	0,8298	0,6415
LDWPSO	3000	9	0,4101	0,8233	0,9483	0,7065
LDWPSO	10000	9	0,4600	0,7860	0,9297	0,6552
LDWPSO	20000	9	0,4361	0,8103	0,9295	0,6925
Promedio			0,4552	0,7874	0,9093	0,6739

**Tabla D.9 Resultados de la evaluación para MLP con QPSO con 5 nodos ocultos con función de activación sigmoïdal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
QPSO	1000	5	0,5959	0,7950	0,8558	0,7292
QPSO	3000	5	0,4304	0,8200	0,9519	0,6958
QPSO	10000	5	0,4571	0,7945	0,9165	0,6850
QPSO	20000	5	0,4512	0,7873	0,8927	0,6899
Promedio			0,4837	0,7992	0,9042	0,7000

**Tabla D.10 Resultados de la evaluación para MLP con LDWQPSO con 4 nodos ocultos con función de activación sigmoïdal para la capa oculta y de salida, utilizando validación cruzada, segundos modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWQPSO	1000	4	0,3782	0,8050	0,9091	0,7030
LDWQPSO	3000	4	0,5347	0,7850	0,9524	0,6242
LDWQPSO	10000	4	0,4446	0,7990	0,9714	0,6341
LDWQPSO	20000	4	0,4519	0,8010	0,9618	0,6420
Promedio			0,4524	0,7975	0,9487	0,6508

**Tabla D.11 Resultados de la evaluación para MLP con LDWPSO con 44 nodos ocultos con función de activación sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWPSO	1000	44	0,5733	0,7400	0,7727	0,7143
LDWPSO	3000	44	0,5483	0,7650	0,9123	0,6096
LDWPSO	10000	44	0,5408	0,7725	0,8501	0,6927
LDWPSO	20000	44	0,5135	0,7998	0,9084	0,6903
Promedio			0,5440	0,7693	0,8609	0,6767

**Tabla D.12 Resultados de la evaluación para MLP con QPSO con 50 nodos ocultos con función de activación sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
QPSO	1000	50	0,5633	0,7500	0,5977	0,8673
QPSO	3000	50	0,5466	0,7667	0,8520	0,6791
QPSO	10000	50	0,5238	0,7895	0,9133	0,6647
QPSO	20000	50	0,5160	0,7973	0,9633	0,6377
Promedio			0,5374	0,7759	0,8316	0,7122

**Tabla D.13 Resultados de la evaluación para MLP con LWDQPSO con 27 nodos ocultos con función de activación sigmoïdal para la capa oculta y softmax para la capa de salida, utilizando validación cruzada, terceros modelos**

Tipo de PSO	Muestra	no	Error	Exactitud	Especificidad	Sensibilidad
LDWQPSO	1000	27	0,5683	0,7450	0,8421	0,6571
LDWQPSO	3000	27	0,6049	0,7083	0,7716	0,6495
LDWQPSO	10000	27	0,5418	0,7715	0,9260	0,6251
LDWQPSO	20000	27	0,5288	0,7845	0,8570	0,7140
Promedio			0,5610	0,7523	0,8492	0,6614