

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

**REDES NEURONALES RECURRENTE CON ALGORITMO DE
OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS PARA
PRONÓSTICO MULTIVARIADO DE CAPTURA DE
ANCHOVETAS EN EL NORTE DE CHILE**

PAMELA ISABEL SEPÚLVEDA PINTO

TESIS DE GRADO
MAGÍSTER EN INGENIERÍA INFORMÁTICA

DICIEMBRE, 2011

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

**REDES NEURONALES RECURRENTE CON ALGORITMO DE
OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS PARA
PRONÓSTICO MULTIVARIADO DE CAPTURA DE
ANCHOVETAS EN EL NORTE DE CHILE**

PAMELA ISABEL SEPÚLVEDA PINTO

Profesor Guía: **Dr. Nivaldo Rodríguez Agurto**

Programa: **Magíster en Ingeniería Informática**

Diciembre, 2011

RESUMEN

Esta tesis de grado está asociada al pronóstico de captura de Anchovetas en la Zona Norte de Chile basado en modelos de Redes Neuronales Recurrentes con aprendizaje PSO (Optimizado por Enjambre de Partículas) y variantes IPSO y DAPSO, disponiendo de variables ambientales como entrada. Las variables externas corresponden a la Temperatura de la superficie del mar (TSM) y Temperatura del mar Región Niño 3.4 (N34). Junto con la variable interna, Captura (CA), forman la entrada multivariada. El proceso de selección de variables aplica algoritmo “mínima Redundancia-Máxima Relevancia” (mRMR) basado en Correlación de Pearson e Información Mutua. La metodología consistió en comenzar con la implementación univariada, para luego incorporar la entrada multivariada logrando complementar estudios anteriores de solución univariada. El mejor pronóstico resultó con $R^2 = 0.86$ correspondiente al modelo con entrada multivariada (CA_{t-1} a $t-6$, TSM_{t-7} , $N34_{t-24}$) y aprendizaje PSO. El segundo mejor es el modelo univariado (CA_{t-1} a $t-6$) de $R^2 = 0.85$ con similar aprendizaje. Las variantes IPSO y DAPSO no alteraron el aprendizaje, poseen rendimientos cercanos al algoritmo PSO en ambas mejores soluciones.

Palabras clave: Anchovetas, Redes Neuronales Recurrentes, Optimización por Enjambre de Partículas (PSO), Entrada Multivariada.

ABSTRACT

This thesis is associated with prognosis of anchovy catch in the northern region of Chile based on Recurrent Neural Network models with learning PSO (Particle Swarm Optimized) and IPSO and DAPSO variants, featuring environmental variables as input. External variables correspond to the surface temperature (TSM) and Sea Temperature Region *Niño 3.4* (N34). Along with the internal variable, Catch (CA), form the multivariate input. The process of variable selection algorithm applies "minimum Redundancy-Maximum Relevance" (mRMR) based on Pearson Correlation and Mutual Information. The methodology was to start implementing univariate and then multivariate achieving incorporate input complement previous studies of univariate solution. The best prognosis was with $R^2 = 0.86$ for Multivariate input (CA_{t-1} a $t-6$, TSM_{t-7} , $N34_{t-24}$) and PSO learning. It's second the univariate model (CA_{t-1} a $t-6$) of $R^2 = 0.85$ with similar learning. The variants DAPSO and IPSO did not alter learning, have performance close to the PSO algorithm in both better solutions.

Keywords: Anchovy, Recurrent Neural Networks, Particle Swarm Optimized (PSO), Entry Multivariate.

CONTENIDO

1	INTRODUCCIÓN	1
2	OBJETIVOS.....	5
2.1	Objetivo General.....	5
2.2	Objetivos Específicos.....	5
3	METODOLOGÍA	6
3.1	Procesos Internos	7
4	ESTADO DEL ARTE.....	8
4.1	Redes Neuronales Artificiales RNA.....	8
4.1.1	Revisión Histórica (11)	8
4.1.2	Red Neuronal Biológica.....	9
4.1.3	Redes Neuronales Artificiales.....	11
4.1.4	Pesos Sinápticos.....	12
4.1.5	Función de Transferencia.....	12
4.1.6	Función de Activación	13
4.2	Tipos de Aprendizaje en RNA.....	15
4.2.1	Aprendizaje Supervisado	15
4.2.2	Modelos de Aprendizaje Supervisado según Valor Deseado	16
4.2.3	Aprendizaje No Supervisado	17
4.3	Tipos de estructuras de RNA.....	18
4.3.1	Redes Monocapa.....	18

4.3.2 Redes Multicapa.....	19
4.4 Redes Neuronales Feedforward y Recurrentes	19
4.4.1 Redes con Propagación hacia adelante (Feedforward)	19
4.4.2 Redes Neuronales Recurrentes.....	21
4.5 Ventajas de las Redes Neuronales Artificiales.....	27
4.6 Algoritmos Evolutivos para Optimización de Parámetros en RNA	28
4.6.1 Inteligencia de Enjambres.....	28
4.6.2 Algoritmo Optimización por Enjambre de Partículas (PSO).....	29
4.7 Indicadores de Bondad del Modelo	33
4.7.1 Coeficiente de Correlación de Pearson (R).....	33
4.7.2 Coeficiente de Determinación (R^2)	34
4.7.3 Error Cuadrático Medio (<i>Mean Squared Error, MSE</i>)	34
4.7.4 Raíz del Error Cuadrático Medio (Root Mean Squared Error, MSE).....	34
4.7.5 Error Medio (Mean Error, ME).....	35
4.7.6 Error Absoluto Medio (<i>Mean Absolute Error, MAE</i>)	35
4.7.7 Error Porcentual Absoluto (<i>Absolute Percentage Error, APE</i>).....	35
4.7.8 Error Porcentual Absoluto Medio (Mean Absolute Percentage Error, MAPE).....	35
4.7.9 Media de las BIAS (<i>MB</i>).....	35
4.7.10 Coeficiente U de Theil (<i>Theil's U, U</i>).....	35
4.8 Selección de Variables.....	36
4.8.1 Métodos de Selección de Variables	36
4.8.2 Métodos Algorítmicos de Selección de Variables explicativas en la Regresión	38
4.8.3 Algoritmo de Selección de mínima Redundancia y Máxima Relevancia (mRMR) .	39
4.8.4 Criterio Coeficiente de Correlación de Pearson (R)	39
4.8.5 Criterio Información Mutua Normalizada (IMN)	39
5 MARCO TEÓRICO DEL TRABAJO.....	41

5.1 Red Neuronal Parcialmente Recurrente de Jordan (RNRJ)	41
5.2 Función de Transferencia.....	44
5.3 Funciones de Activación.....	44
5.4 Aprendizaje Supervisado con Recurrencia de Jordan	44
5.5 Suavizado con Promedio Móvil.....	45
5.6 Algoritmo PSO.....	45
5.6.1 PSO Convencional	45
5.6.2 PSO con Peso de Inercia Dinámico (IPSO)	45
5.6.3 PSO con Adaptación Dinámica (DAPSO).....	46
5.6.4 Dimensión de la Partícula	48
5.7 Función de Costo	49
5.8 Indicadores de Bondad del Modelo	49
5.9 Algoritmo de Selección de Variables	50
5.9.1 Criterio de Selección: Correlación de Pearson (R)	51
5.9.2 Criterio de Selección: Información Mutua Normalizada (IMN).....	51
5.9.3 Algoritmo mínima Redundancia-Máxima Relevancia (mRMR).....	53
6 IMPLEMENTACIÓN.....	56
6.1 Preparación de datos	56
6.1.1 Datos fuente	56
6.1.2 Normalizado Z-score.....	56
6.1.3 Suavizado con Promedio Móvil 3 (PM3)	58
6.2 Experimentación	59
6.3 Entrenamiento y Validación	59

7	RESULTADOS EXPERIMENTALES	60
7.1	Resultado Experimental Solución Univariada	60
7.1.1	Parámetros de Experimentación.....	60
7.1.2	Resultado Solución Univariada.....	61
7.1.3	Mejor resultado Solución Univariada	61
7.1.4	Resultado Regresión Lineal con PseudoInversa	62
7.1.5	Gráfica de Correlación mejor Resultado Solución Univariada.....	62
7.2	Resultado Experimental Solución Multivariada.....	63
7.2.1	Selección de Variables	63
7.3	Resultados Modelo Multivariado	67
7.4	Resultado Experimental Solución Multivariada.....	67
7.4.1	Parámetros de Experimentación.....	67
7.4.2	Resultado Solución Multivariada.....	67
7.4.3	Mejor resultado Solución Multivariada	69
7.4.4	Comparativo de Resultados Soluciones Univariada y Multivariada	69
7.4.5	Comparativo de Resultados de Aprendizajes PSO, IPSO, DAPSO	70
7.4.6	Gráficas del Mejor Resultado Solución Multivariada.....	70
8	CONCLUSIONES	71
9	BIBLIOGRAFÍA.....	73

ÍNDICE DE FIGURAS

Figura 1.1 Región 3.4 del Océano Pacífico Tropical (Ecuatorial).	3
Figura 1.2 Variable Interna CA: Muestras mensuales 1963 - 2006 (44 años, 528 datos).	4
Figura 1.3 Variables Externas TSM y N34: Muestras mensuales 1963 - 2006 (44 años, 528 datos).	4
Figura 3.1 Flujo de procesos del desarrollo metodológico.	6
Figura 4.1. Neurona Biológica.	10
Figura 4.2. Modelo de Neurona Artificial	11
Figura 4.3. Aprendizaje Supervisado.....	15
Figura 4.4. Entrada Cruda - Valor Deseado Crudo.....	16
Figura 4.5. Entrada Suavizada - Valor Deseado Suavizado.....	16
Figura 4.6. Entrada Suavizada - Valor Deseado Crudo.	17
Figura 4.7: Aprendizaje No Supervisado.....	17
Figura 4.8. Red Neuronal Monocapa.....	18
Figura 4.9: Red Neuronal Multicapa	19
Figura 4.10. Red Neuronal Feedforward con una capa oculta y una salida.	20
Figura 4.11. Tipos de Recurrencias	21
Figura 4.12. Red Completamente Recurrente.....	23
Figura 4.13: Red Completamente Recurrente Reorganizada en Capas.....	24
Figura 4.14. Esquemas de Redes Parcialmente Recurrentes.....	25
Figura 4.15. Red de Jordan.....	26
Figura 4.16. Red de Elman.....	27
Figura 4.17: Flujo del Algoritmo PSO.	30
Figura 4.18: Esquema de Movimiento de la Partícula.	32
Figura 4.19: Representaciones de R	33

Figura 4.20. Método de Filtro.....	36
Figura 4.21. Método de Wrapper.....	37
Figura 4.22. Método Embebido.....	37
Figura 5.1 Red Parcialmente Recurrente de Jordan con una salida, sin capa de retardo.....	41
Figura 5.2 Red de Jordan con una salida, sin capa de retardo y sin retención de pasado.....	43
Figura 5.3 Red Recurrente con Entrada Suavizada - Valor Deseado Suavizado.....	44
Figura 5.4 Proceso general Método de Filtro.....	50
Figura 5.5 Gráfico de Ejemplo de Umbral de Relevancia.....	53
Figura 5.6 Gráfico de Ejemplo de Umbral de Redundancia.....	54
Figura 5.7 Algoritmo mRMR.....	55
Figura 6.1 Normalización Z-score de datos fuente de variables CA, TSM y N34.....	57
Figura 6.2 Suavizado PM3 de variables normalizadas CA, TSM y N34.....	58
Figura 7.1 Correlación del mejor resultado univariado.....	62
Figura 7.2 Gráfico de Relevancias de variables CA, TSM y N34 con Correlación de Pearson.....	63
Figura 7.3 Gráfico de Relevancias de variables CA, TSM y N34 con IMN.....	65
Figura 7.4 Gráfica comparativa mejor resultado general (Munivariado).....	70
Figura 7.5 Correlación del mejor resultado.....	70

ÍNDICE DE TABLAS

Tabla 3.1 Descripción del desarrollo metodológico.	6
Tabla 4.1. Funciones de Activación más utilizadas.	14
Tabla 5.1 Funciones de Activación a utilizar en el trabajo.	44
Tabla 5.2 Ejemplo de Matriz Cruzada de Correlación Lineal.	53
Tabla 5.3 Ejemplo de correlación de variables redundantes	54
Tabla 6.1 Variables Ambientales del pronóstico	56
Tabla 6.2 Experimento.	59
Tabla 7.1 Métricas para evaluar el resultado experimental.	60
Tabla 7.2 Parámetros Experimentación Solución Univariada.	60
Tabla 7.3 Ranking Resultado Experimentación Solución Univariada con R^2 superior a 0.84 (inclusive). 61	
Tabla 7.4 Indicadores de Bondad de los mejores resultados de Experimentación Solución Univariada. ...	61
Tabla 7.5 Mejor Resultado Experimentación Solución Univariada.	61
Tabla 7.6 Indicadores de Bondad Solución con PseudoInversa.	62
Tabla 7.7 Parámetros Selección de Variables con criterio Correlación de Pearson.	63
Tabla 7.8 Resultado Selección de Variables con mRMR, criterio Correlación de Pearson.	64
Tabla 7.9 Variables seleccionadas con Correlación de Pearson.	64
Tabla 7.10 Parámetros Selección de Variables con criterio Correlación de Pearson.	65
Tabla 7.11 Resultado Selección de Variables con mRMR, criterio IMN.	66
Tabla 7.12 Variables seleccionadas con IMN.	66
Tabla 7.13 Parámetros Experimentación Solución Multivariada.	67
Tabla 7.14 Experimentación Solución Multivariada.	68
Tabla 7.15 Ranking Resultado Experimentación Solución Integrada con R^2 superior a 0.84.	68
Tabla 7.16 Indicadores de Bondad de los mejores resultados Experimentación Solución Multivariada. ...	69

Tabla 7.17 Mejor Resultado Experimentación Solución Multivariada.	69
Tabla 7.18 Comparativo Resultados Solución Univariado y Multivariado.....	69
Tabla 7.19 Comparativo Resultados de Aprendizajes PSO, IPSO y DAPSO.	70
Tabla 7.20 Ganancia mejor aprendizaje PSO con respecto a variantes IPSO y DAPSO.	70

1 INTRODUCCIÓN

Para satisfacer sus necesidades de alimentación, el hombre, desde tiempo inmemorial, ha realizado la pesca, la cual se ha ido desarrollando con la construcción de instrumentos de captura hasta transformarse en una actividad productiva. Entre las causas que han permitido el mayor aprovechamiento de los recursos pesqueros está en primer lugar la realización de grandes investigaciones oceanográficas y pesqueras que aportaron los datos para que el proceso de la pesca se pudiera industrializar. Hoy en día se continúan los esfuerzos para contribuir en mejorar los procesos de captura de especies y así satisfacer el consumo humano que muestra un ritmo acelerado de crecimiento.

Nuestro país, hoy se encuentra entre los principales productores pesqueros a nivel mundial y por ello requiere contar con elementos de gestión que permitan diseñar mejores políticas y determinar cuotas de pesca. Asimismo, la industria pesquera busca optimizar sus estrategias de pesca considerando las variables ambientales que afectan los recursos y su captura. Lo anterior, determina la necesidad de apoyos científicos que generen información predictiva de la variabilidad del comportamiento de las especies.

La tarea de la investigación apoyada por las técnicas científicas se enfoca en el análisis de especies específicas o de comportamiento muy similar. En la zona nortina del país, las especies *Sardina* (*Sardinops sagax*) y Anchoveta (*Engraulis ringens*) son los principales recursos de interés para su captura y proceso, poseen variaciones imprevistas de biomasa y alternancia de abundancia entre ellas. En este trabajo se analiza la unidad de pesquería pelágica: Anchoveta (1).

La captura de Anchoveta en años anteriores ha sido destinada básicamente al consumo humano indirecto (en la elaboración de harina y aceite), sin embargo la producción de conservas elaboradas a base de esta especie ha incrementado su demanda. De acuerdo a registros de los movimientos productivos de esta especie pelágica del norte de Chile, entre Arica y Antofagasta, se visualiza que mantienen un comportamiento no lineal (sus cambios no se basan en una simple relación proporcional entre causa y efecto, éstos suelen ser bruscos, inesperados y difíciles de prever) y no estacionario (sus propiedades varían con el tiempo), pues reflejan notables variaciones de biomasa, rápidos colapsos y recuperaciones, asociadas a cambios ambientales como la temperatura superficial del mar y el fenómeno de El Niño Oscilación del Sur” (ENOS) (conocido como “Corriente del Niño”).

Actualmente, existen estudios predictivos que procesan datos para comportamientos lineales, por lo tanto la escasez de aplicaciones de pronóstico de comportamientos no lineales impulsa la necesidad de esta experiencia de estudio.

Ante la complejidad señalada, la aplicación del análisis de pronóstico por Redes Neuronales satisface tal comportamiento. Las Redes Neuronales Artificiales (RNA), que emulan las características propias del ser humano como la capacidad de memorizar y de asociar, se materializan a través de “modelos” que poseen diferentes topologías y técnicas de entrenamiento según el problema en particular.

El presente estudio trabaja sobre modelos basados en Redes Neuronales Recurrentes con Algoritmo de Aprendizaje denominado Optimización por Enjambre de Partículas (Particle Swarm Optimization - PSO) cuyo enfoque está inspirado en el comportamiento social de individuos. Considerando que el objetivo es pronosticar la variabilidad de abundancia de una especie pelágica afectada por variables ambientales, estos modelos reciben como entrada aquellas variables más informativas a partir de un proceso previo de selección, que mejoran el procesamiento de los datos sin alterar la representación original de los atributos (12).

La bondad de las soluciones es evaluada con métricas de calidad tales como el Coeficiente de Determinación (R^2), Error Cuadrático Medio (MSE), Error Porcentual Absoluto (APE) y Error Porcentual Absoluto Medio ($MAPE$), entre otras. En base a estas estadísticas se puede seleccionar la topología neuronal del modelo que explique mejor la variabilidad del comportamiento del sistema (19).

ANCHOVETA (ENGRAULIS RINGENS)



La Anchoveta (Engraulis Ringens) es una especie pelágica perteneciente a la familia de los Clupeidos, ubicada en el género Engraulius. Es una de las especies pelágicas de mayor importancia debido a los grandes volúmenes de captura anual en el ámbito mundial. Viven en aguas oceánicas cuya temperatura se encuentra entre 14,5 y los 20°C. La anchoveta vive hasta los 3 o 4 años de edad y en su etapa adulta, alcanza una longitud que oscila entre 12 y 16 centímetros. Una hembra adulta produce millares de huevos durante su vida, desovando en la superficie y hasta 50 metros de profundidad.

Las series de datos para el estudio provienen de registros históricos del Servicio Nacional de Pesca (www.sernapesca.cl) (20), corresponden a 44 años que se extienden desde 1963 a 2006 y equivalen a 528 muestras mensuales de cada variable ambiental a considerar. La variable interna, dependiente, corresponde a **Captura de Anchoveta (CA)** en la zona norte del país (Figura 1.2). Las variables externas, independientes, corresponden a **Temperatura Superficial del Mar (TSM)** y **Temperatura del mar de la Región Niño 3.4 (N34)** ubicada en el Océano Pacífico Tropical (Ecuatorial) (Figura 1.3).

La variable oceánica asociada a la temperatura superficial del mar, es monitoreada por el Servicio Hidrográfico y Oceanográfico de la Armada de Chile (SHOA) (www.shoa.cl) a través de una red de estaciones a nivel del mar, localizados a lo largo de la costa nacional. Entre los años 1950 y 1998 la temperatura se registraba manualmente utilizando un termómetro de mercurio. En la actualidad se cuenta con un sistema automático de registro con termistores que transmiten en tiempo casi real vía satelital. Los datos de TSM son enviados mensualmente al Boletín de Alerta Climática (BAC) de la Comisión permanente del Pacífico Sur (CPPS) (<http://www.inocar.mil.ec/index.php>).

La variable Temperatura del mar de la Región Niño 3.4 está asociada al fenómeno de El Niño Oscilación del Sur (ENOS). "El Niño" o la "Corriente del Niño", es un ciclo oceánico y atmosférico que ocurre cada cierto intervalo de años, debido a cambios en las condiciones del viento que ocurren en la zona del Ecuador. El movimiento de las aguas cálidas hacia Australia se desplaza lentamente de regreso a la zona del Ecuador y ese ejercicio natural, finalmente, genera una temperatura superior entre 3 y 7 grados más que lo normal hacia el sur y al norte pegado a la costa Pacífica de América. Por lo anterior, una de las variables de mayor importancia en el monitoreo del Fenómeno del Niño es la temperatura de la Superficie del Mar en el Océano Pacífico Tropical (Ecuatorial). Para esto el Pacífico central se ha dividido en 4 regiones, la zona de interés para este estudio está entre las regiones 3 y 4, se denomina Región 3.4 y está localizada entre las coordenadas 5°N–5°S y 170°–120°W (1) (Figura 1.1)(www.hidromet.com.pa).

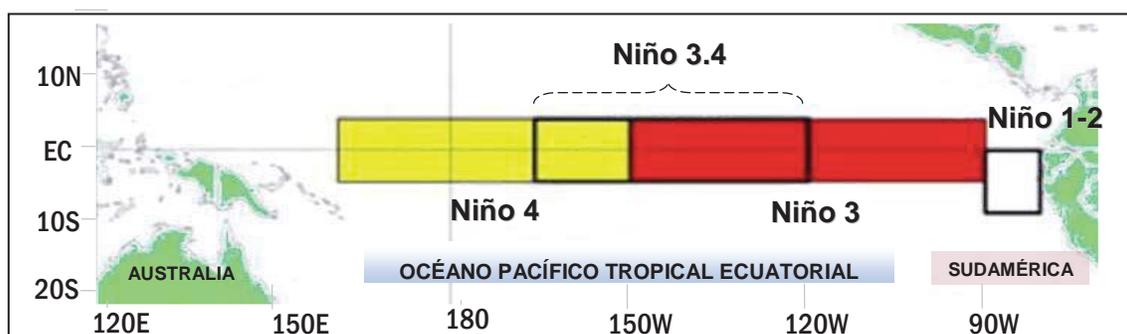


Figura 1.1 Región 3.4 del Océano Pacífico Tropical (Ecuatorial).

La variable CA está medida en toneladas y las variables externas TSM y N34 en grado Celsius (C°).

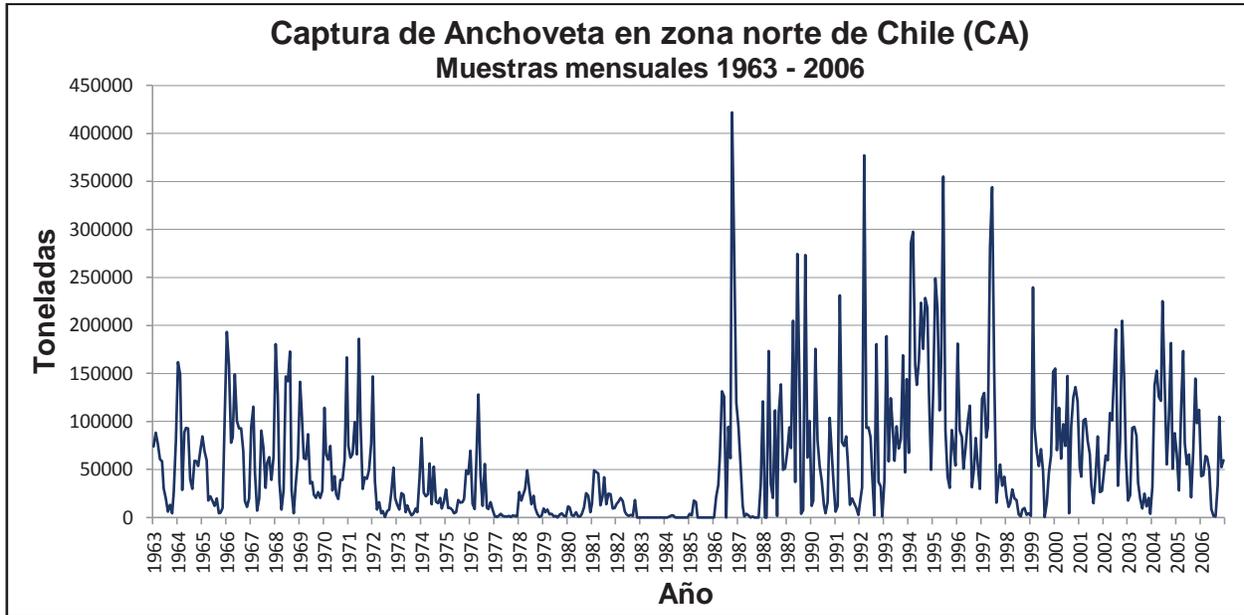


Figura 1.2 Variable Interna CA: Muestras mensuales 1963 - 2006 (44 años, 528 datos).

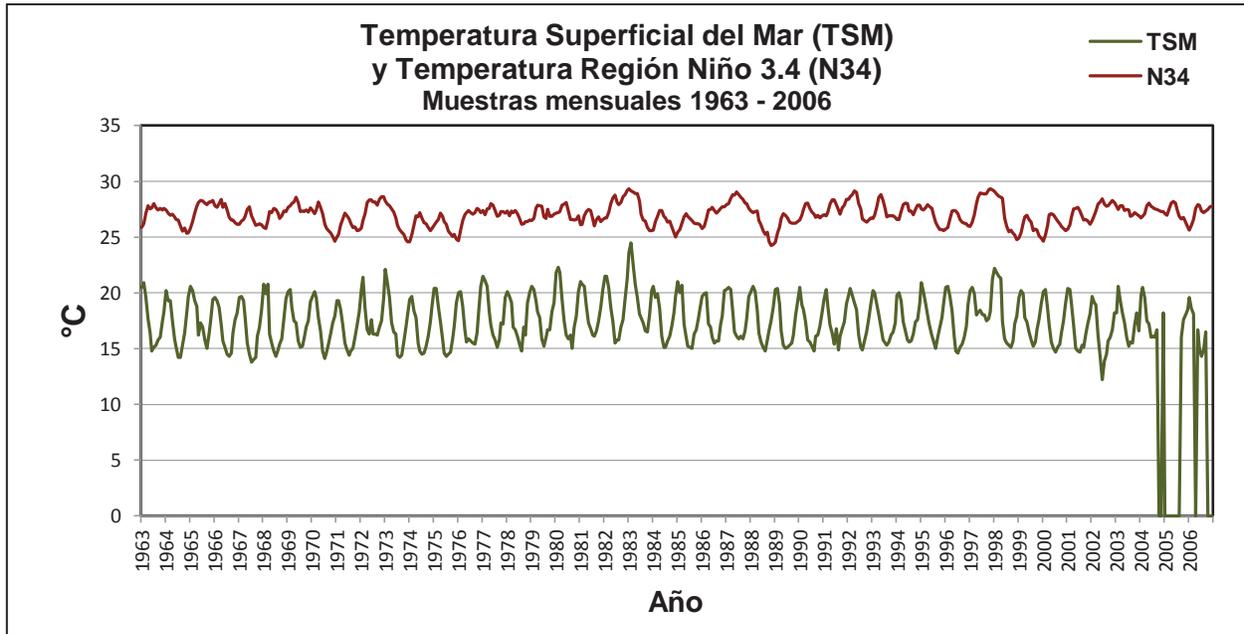


Figura 1.3 Variables Externas TSM y N34: Muestras mensuales 1963 - 2006 (44 años, 528 datos).

2 OBJETIVOS

2.1 Objetivo General

Desarrollar y evaluar un modelo de pronóstico del comportamiento de biomasa de Anchovetas en la zona norte del país de Chile (entre Arica y Antofagasta) basado en Redes Neuronales Recurrentes con entrada multivariada y Algoritmos de Aprendizaje (Optimización) por Enjambre de Partículas (PSO) y sus variantes PSO con Peso de Inercia Dinámico (IPSO) y PSO con Adaptación Dinámica (DAPSO).

2.2 Objetivos Específicos

- Estudiar y analizar Modelos Neuronales Recurrentes con Algoritmos de Aprendizaje PSO convencional y sus variantes IPSO y DAPSO.
- Estudiar Métodos de Selección de Variables para reducir la dimensionalidad de entradas redundantes o con información irrelevante.
- Implementar y evaluar los modelos neuronales con data suavizada para solución univariada y multivariada.

3 METODOLOGÍA

La metodología se basa en dos etapas que se desarrollan en forma secuencial y con procesos internos similares:

Etapa 1: Desarrollo de Solución Univariada		Etapa 2: Desarrollo de Solución Multivariada	
Entrada	Aprendizaje	Entrada	Aprendizaje
CA	PSO, IPSO, DAPSO	CA, TSM, N34	PSO, IPSO, DAPSO

Tabla 3.1 Descripción del desarrollo metodológico.

De la primera etapa, entrada univariada con variable Captura (CA) y aprendizajes PSO, IPSO y DAPSO, se obtienen resultados parciales de la topología del modelo neuronal que son considerados en la fase posterior. La segunda etapa incorpora al modelo dos variables externas: Temperatura Superficial del Mar (TSM) y Temperatura Región Niño 3.4 (N34) conformando la entrada multivariada para evaluar mejoras de calidad.

El objeto de esta metodología es comparar los resultados predictivos de soluciones de diferente complejidad y con ciertos parámetros comunes en la topología de sus modelos neuronales. El proceso integral cumple con la siguiente cadena de valor:

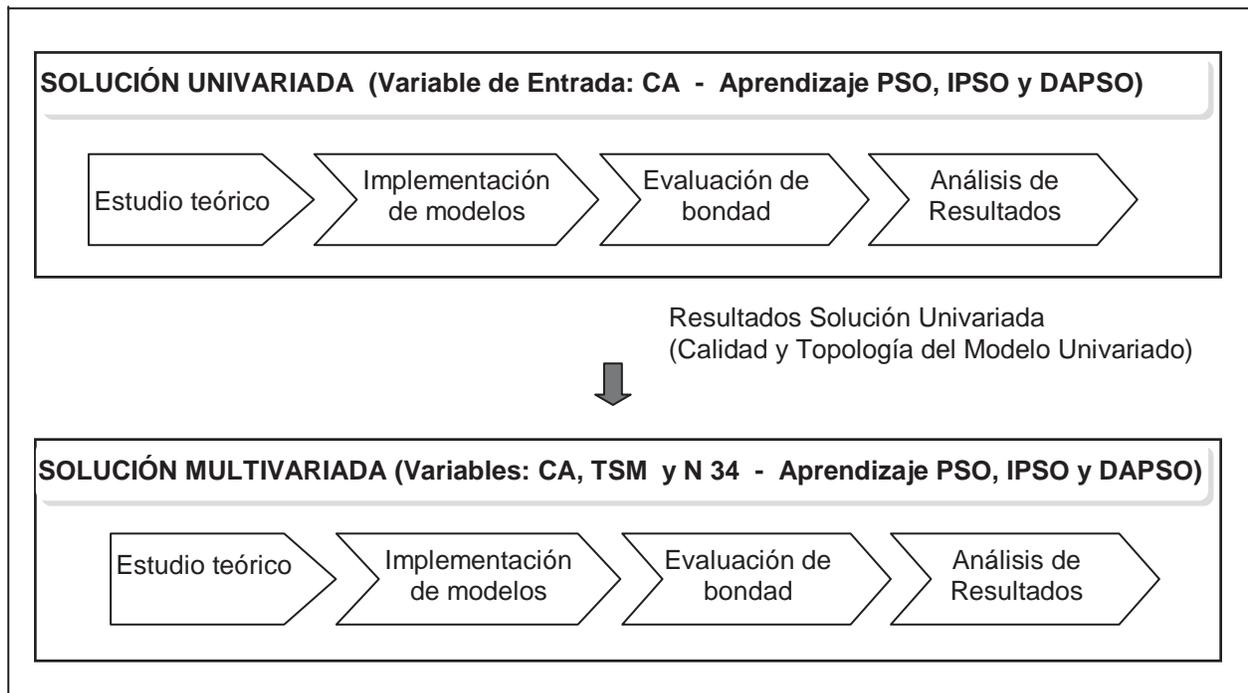


Figura 3.1 Flujo de procesos del desarrollo metodológico.

3.1 Procesos Internos

Estudio Teórico Etapa 1 Solución Univariada

- Estudio de Redes Neuronales Artificiales (RN).
- Estudio de Redes Neuronales Recurrentes (RNR).
- Estudio de Algoritmos basados en Optimización por Enjambre de Partículas Convencional (PSO) y variantes IPSO y DAPSO.
- Estudio de preparación de datos de entrada para modelo RNR.
- Estudio de métricas para evaluar la bondad del modelo predictivo (Calidad).

Estudio Teórico Etapa 2 Solución Multivariada

- Estudio de Métodos de Selección de Variables para modelos multivariados.
- Estudio de Selección de Variables con filtro basado en algoritmo mínima Redundancia-Máxima Redundancia (mRMR)
- Estudio de Selección de Variables con filtro basado en Información Mutua Normalizada (NMI)

Implementación de Modelos Etapa 1 y Etapa 2

- Desarrollo de los algoritmos asociados a la etapa en desarrollo.
- Preparación de datos aplicando suavizado (Promedio Móvil 3) y normalización.
- En Etapa 2 de solución multivariada, aplicación de filtro de variables por medio de algoritmo mRMR con métricas Correlación Pearson e Información Mutua.

Evaluación de bondad Etapa 1 y Etapa 2

- Ejecución de experimentos aplicando métricas de calidad.

Análisis de Resultados Etapa 1 y Etapa 2

- Identificación de la topología del modelo neuronal mejor evaluado.
- En Etapa 2 de solución multivariada, análisis comparativo con primera etapa.

4 ESTADO DEL ARTE

4.1 Redes Neuronales Artificiales RNA

4.1.1 Revisión Histórica (11)

- En **1943**, Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, fueron los primeros teóricos que concibieron los fundamentos de la computación neuronal y modelaron una red neuronal simple mediante circuitos eléctricos, en base a su teoría acerca de cómo trabajan las neuronas.
- En **1949**, Donald Hebb, neuropsicólogo, fue el primero en explicar los procesos del aprendizaje desde un punto de vista fisiológico, postulando formalmente que el aprendizaje estaba basado en la modificación de las conexiones sinápticas entre neuronas, es decir, cuando son activados ciertos cambios en una neurona. Hoy, éste sigue siendo el fundamento de la mayoría de las funciones de aprendizaje que pueden hallarse en una red neuronal.
- En **1951**, Marvin Minsky construye el primer neurocomputador (Snark). Resulta un éxito desde el punto de vista técnico, pues ajusta automáticamente sus pesos, pero no se llega a darle ninguna aplicación práctica.
- En **1957**, Frank Rosenblatt, Charles Whightman y otros, desarrollaron el Perceptrón, que es la red neuronal artificial más antigua. Aunque tenía una serie de limitaciones, este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado anteriormente.
- En **1960**, Bernard Widrow y Marcial Hoff desarrollaron el modelo Adaline (ADaptative LINear Elements), quea fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) que se ha utilizado comercialmente durante varias décadas.
- En **1969**, Minsky y Papert publican "Perceptrons", donde se prueba que un perceptrón no puede resolver problemas linealmente no separables, argumento que sirve para que se lleve a cabo una verdadera campaña anti redes neuronales artificiales, lo que debilita notoriamente la investigación en el área.

- En **1974**, Paul Werbos ideó el principio matemático que permitió desarrollar el algoritmo de aprendizaje de propagación hacia atrás (retro-propagación o backpropagation); propuesto posterior e independientemente por Rumelhart, Hinton y Williams como un método para entrenar una red de perceptrones de varios niveles.
- En **1980**, Kunihiko Fukushima planteó un modelo neuronal para el reconocimiento de patrones visuales. Algunos años después, en 1985, John Hopfield, físico de gran reputación, inició el renacimiento de la computación neuronal, dictando conferencias y publicando el libro "Computación neuronal de decisiones en problemas de optimización".
- En **1986**, David Rumelhart y Geoffrey Hinton replantearon el algoritmo de aprendizaje de propagación hacia atrás (backpropagation) y, a partir de ese año -y hasta hoy- las investigaciones y el desarrollo de las redes neuronales se popularizaron, especialmente para aplicaciones nuevas las áreas de control y simulación.

4.1.2 Red Neuronal Biológica

El cerebro es un computador complejo, no lineal y paralelo. Tiene la capacidad para realizar tareas tales como el reconocimiento de patrones, la percepción y el control motor, mucho más rápido que cualquier computador, a pesar de que los eventos ocurren en el rango de los milisegundos en los sistemas neuronales, contra los nanosegundos en los componentes de silicio. Además de estas características, otras, como la capacidad de aprender, memorizar e, incluso, generalizar, han promovido la investigación en el modelamiento algorítmico de sistemas neuronales biológicos, denominados redes neuronales artificiales (RNA). Se estima que hay del orden de 10-500 mil millones de neuronas en la corteza humana, con 60 mil millones de sinapsis. Las neuronas se organizan en aproximadamente 1000 módulos principales, cada uno con cerca de 500 redes neuronales. Por ello, por el momento no es posible modelar el cerebro humano. Los éxitos actuales en el modelado neural ocurren con pequeñas RNAs artificiales resolviendo alguna tarea específica(3). Con RNAs de tamaño medio se pueden resolver problemas con un único objetivo con bastante facilidad, limitado por la capacidad de potencia de cómputo y espacio de almacenamiento disponibles actualmente. El cerebro tiene, sin embargo, la capacidad de resolver varios problemas al mismo tiempo utilizando piezas distribuidas.

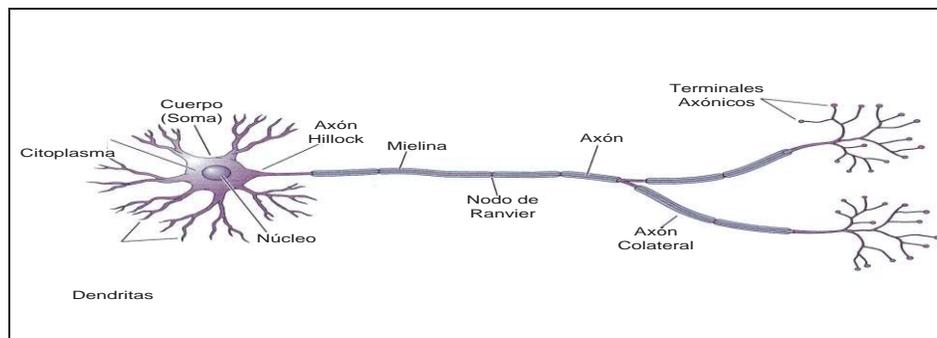


Figura 4.1. Neurona Biológica.

Los elementos básicos de los sistemas biológicos neuronales son las células nerviosas denominadas neuronas. Una neurona es una célula y, como tal, posee todos los elementos comunes de las células biológicas. A su vez, tienen características propias que le permiten comunicarse entre ellas, lo que las diferencia del resto de las células biológicas. Como se ilustra en la Figura 4.1, una neurona consta de un cuerpo celular, dendritas y un axón. Las neuronas están masivamente interconectadas, donde la interconexión es entre el axón de una neurona y una dendrita de otra. Esta conexión se conoce como sinapsis. Las señales se propagan a partir de las dendritas, a través del cuerpo celular al axón, de donde las señales se propagan a todas las dendritas conectadas. Una señal se transmite a lo largo del axón de una neurona sólo cuando la célula "dispara". Una neurona puede tanto inhibir como excitar una señal. En la corteza cerebral se observa una organización horizontal en capas y una organización vertical en columnas de neuronas. La intensidad de una sinapsis no es fija, sino que puede ser modificada en base a la información proveniente del medio. De esta manera la estructura del cerebro no permanece fija sino que se va modificando por la formación de nuevas conexiones, ya sean excitadoras o inhibitorias, la destrucción de conexiones, la modificación de la intensidad de la sinapsis, o incluso por muerte neuronal.

La comunicación entre neuronas ocurre cuando en el núcleo de las neuronas transmisoras o pre-sinápticas se genera un pulso eléctrico llamado potencial de acción. El pulso eléctrico se propaga a través del axón en dirección a las sinapsis. La información se transmite a las neuronas vecinas utilizando un proceso químico, mediante la liberación de neurotransmisores. Estos neurotransmisores se transmiten a través de la sinapsis hacia la neurona receptora. La neurona receptora o post-sináptica toma la señal enviada por cientos de neuronas a través de las dendritas y la transmite al cuerpo celular. Estas señales pueden ser excitadoras (positivas) o inhibitorias (negativas) (4). El núcleo es el encargado de integrar la información proveniente de las distintas neuronas. Si la señal resultante supera un determinado umbral (umbral de disparo) el soma emite un pulso que se transmite a lo largo del axón dando lugar a la transmisión eléctrica a lo largo de la neurona. Al llegar la señal al extremo del axón se liberan neurotransmisores que permiten transmitir la señal a las neuronas vecinas.

4.1.3 Redes Neuronales Artificiales

Las Redes Neuronales Artificial (RNA) son sistemas que hacen uso los principios conocidos de la organización del cerebro humano. Consisten en un número de procesadores independientes y simples: las neuronas, cuya función es determinada por la estructura de la red. Estas neuronas, conectadas en paralelo, se comunican con otras por medio de la fuerza de las conexiones ponderadas, los pesos sinápticos. Cada neurona acumula la entrada que recibe y genera una salida de acuerdo a una función de activación interna.

Una neurona artificial puede ser modelada con tres elementos básicos (5):

- Una serie de sinapsis o entradas, cada una está caracterizada por un peso que puede tomar tanto valores negativos como positivos.
- Una función de transferencia, usualmente un sumador o multiplicador, que procesa las señales de entrada hacia la neurona, de acuerdo a sus peso sinápticos.
- Una función de activación para limitar la amplitud de la salida de la neurona. Ésta corresponde habitualmente a una función no lineal que normaliza el estado interno de la neurona a un intervalo cerrado $[0,1]$ o alternativamente $[-1,1]$. Ejemplos de función de activación son: la función escalón, sigmoidea, gaussiana, polinomial, entre otras.

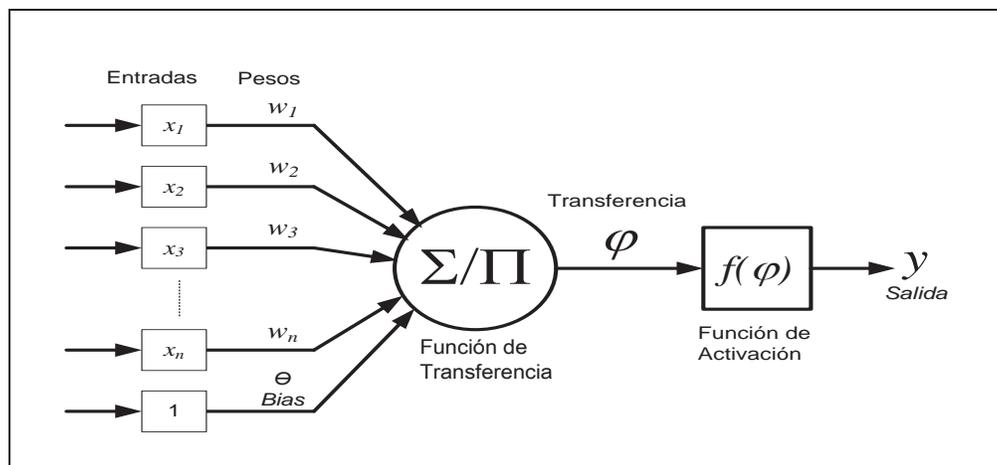


Figura 4.2. Modelo de Neurona Artificial

4.1.4 Pesos Sinápticos

El peso sináptico w_{ij} define la fuerza de una conexión sináptica entre dos neuronas, la neurona pre-sináptica i y la neurona post-sináptica j . Los pesos sinápticos pueden tomar valores positivos, negativos o cero. En caso de una entrada positiva, un peso positivo actúa como excitador, mientras que un peso negativo actúa como inhibidor. En caso de que el peso sea cero, no existe comunicación entre el par de neuronas. Mediante el ajuste de los pesos sinápticos la red es capaz de adaptarse a cualquier entorno y realizar una determinada tarea.

Las redes neuronales pueden tener factores de peso adaptable o fijo. Las que tienen pesos adaptables emplean leyes de aprendizaje para ajustar el valor de la fuerza de interconexión con otras neuronas. Si se utilizan pesos fijos, su tarea debe estar previamente definida. Los pesos son determinados a partir de una descripción completa del problema a tratar. Los pesos adaptables son muy importantes, ya que no se conoce de antemano su valor correcto.

Cada peso es una memoria de la experiencia pasada de la neurona, la cual es capaz de adaptarse a nuevas experiencias a través del aprendizaje.

4.1.5 Función de Transferencia

La señal de entrada (pre-sináptica) a una Neurona Artificial es el resultado global de la aplicación de una función de transferencia sobre las señales de entrada a la red neuronal. También llamada regla de propagación, determina el potencial resultante de la interacción de la neurona j con las N neuronas vecinas que la preceden. El potencial resultante (transferencia) φ_j se puede expresar de la siguiente manera (6):

$$\varphi_j = \phi_j(w_{ij}, x_i, \theta) \quad (1)$$

donde φ_j es la señal de entrada a la neurona j , w_{ij} los pesos sinápticos que llegan a la neurona j desde cada entrada x_i .

Las señales de entrada netas a las neuronas son de diversos tipos, tales como unidades de producto (product units) o unidades de agregación (summation units) en el caso de las redes feedforward (7). Otros tipos de topologías de red usan diferentes tipos de señales de entrada neta, como los mapas auto-organizados que usan métricas de distancia como la distancia euclidiana. La elección del tipo de entrada neta influye en la salida de la red, los algoritmos de entrenamiento, los algoritmos de selección de arquitectura, el tipo de datos que puede procesar una neurona, así como la elección de la función de activación.

Las entradas y salidas de una neurona pueden ser clasificadas en dos grandes grupos, binarias o continuas (6). Las neuronas binarias (digitales) sólo admiten dos valores posibles. En general en este tipo de neurona se utilizan los siguientes dos alfabetos $\{0,1\}$ o $\{-1,1\}$. Por su parte, las neuronas continuas (analógicas) admiten valores dentro de un determinado rango, que en general suele definirse como $[-1, 1]$. La selección del tipo de neurona a utilizar depende de la aplicación y del modelo a construir.

4.1.6 Función de Activación

La función de activación, recibe las señales de entrada y las bias de la red, a través de la función de transferencia, y determina la salida (o la fuerza de disparo) de la neurona.

También llamada función de salida, proporciona el valor de salida desde la neurona, en base a su valor de transferencia.

Se pueden usar diferentes tipos de funciones de activación. En general, las funciones de activación son monótonamente crecientes, donde (con exclusión de la función de identidad) se da:

$$f(-\infty) = 0 \quad \text{o} \quad f(-\infty) = -1 \quad (2)$$

y

$$f(\infty) = 1 \quad (3)$$

La función de activación determina el estado de activación actual de la neurona en base al potencial resultante de la transferencia φ_i y al estado de activación anterior de la neurona $a_i(t-1)$. El estado de activación de la neurona para un determinado instante de tiempo t puede ser expresado de la siguiente manera:

$$a_i(t) = f_i(a_i(t-1), \varphi_i(t)) \quad (4)$$

Sin embargo, en la mayoría de los modelos se suele ignorar el estado anterior de la neurona, definiéndose el estado de activación en función del potencial resultante φ_i :

$$a_i(t) = f_i(\varphi_i(t)) \quad (5)$$

La tabla siguiente muestra las funciones de activación más utilizadas en los distintos modelos de redes neuronales artificiales.

Función de Activación	Fórmula	Rango
Identidad	$f(x) = x$	$[-\infty, \infty]$
Escalón	$f(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$	$[0,1]$
	$f(x) = \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$	$[-1,1]$
Lineal a Tramos	$f(x) = \begin{cases} x & \text{si } -1 \leq x \leq 1 \\ +1 & \text{si } x > 1 \\ -1 & \text{si } x < -1 \end{cases}$	$[-1,1]$
Sigmoidal (Logística)	$f(x) = \frac{1}{1 + e^{-x}}$	$[0,1]$
	$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$[-1,1]$
Gaussiana	$f(x) = e^{-x^2}$	$[0,1]$
Sinusoidal	$f(x) = \text{sen}(\pi x)$	$[-1,1]$

Tabla 4.1. Funciones de Activación más utilizadas.

La función de activación cumple con el objetivo de limitar el rango de salida de la neurona y puede ser lineal o no lineal (4). La función logística es la más comúnmente usada por sus características de derivación y se recomienda para problemas de predicción (aunque en algoritmos de retro-propagación la función más comúnmente utilizada es la logística, éstos pueden trabajar con cualquier otra función de activación que sea diferenciable. Una buena función de activación debería cumplir: primero con que ella misma y su derivada sean fáciles de computar y segundo, que la función debe tener una amplia parte lineal para lograr velocidad de entrenamiento y convergencia en pocos ciclos. Precisamente la función logística (logistic sigmoidal) es usada comúnmente porque su derivada es una de las más fáciles de computar(4).

Umbral

El umbral interno del nodo (θ) es la magnitud del desplazamiento que afecta a la activación de la salida y del nodo, de la siguiente manera:

Las unidades sigmoidales ocultas y de salida usualmente utilizan un término de “umbral” o “bias” en el cómputo de la señal de entrada neta a la neurona. En una unidad de salida lineal, la bias es equivalente a la intercepción en un modelo de regresión.

La bias puede ser tratada como un peso de conexión desde una unidad especial con un valor de señal constante, distinto de cero. El término “bias” es utilizado habitualmente para referirse a una “unidad bias” con un valor constante de 1. No todos los autores siguen esta distinción. Independientemente de la terminología, si las bias o umbrales se suman o restan no tiene ningún efecto sobre el desempeño de la red

La unidad de bias se conecta a cada neurona oculta o de salida que necesite un término de bias. Por lo tanto, los términos de bias pueden ser aprendidos de la misma manera que los otros pesos.

4.2 Tipos de Aprendizaje en RNA

En el proceso de entrenamiento de las redes neuronales, se utilizan básicamente tres tipos de aprendizaje: Supervisado, No Supervisado y Por Refuerzo. El aprendizaje consiste en ajustar los valores de pesos y umbrales hasta que se satisfagan ciertos criterios.

4.2.1 Aprendizaje Supervisado

Se le proporciona a la red tanto la *salida* como la *entrada correcta*, y la red ajusta sus pesos para disminuir el error en la salida que ella calcula. Estos datos se denominan “de entrenamiento”. En general, el aprendizaje supervisado se basa en la presentación sucesiva de los datos de entrada X y comparación de la respuesta de la red con los valores Y reales observados. Se mide la distancia entre respuesta y observaciones y se modifican los parámetros de la red hasta que la distancia se hace mínima. Cada vector de aprendizaje X_i, Y_i (cada dato) es sometido a la red de forma secuencial. La secuencia se repite muchas de veces hasta que la red se estabiliza y deja de sufrir modificaciones.

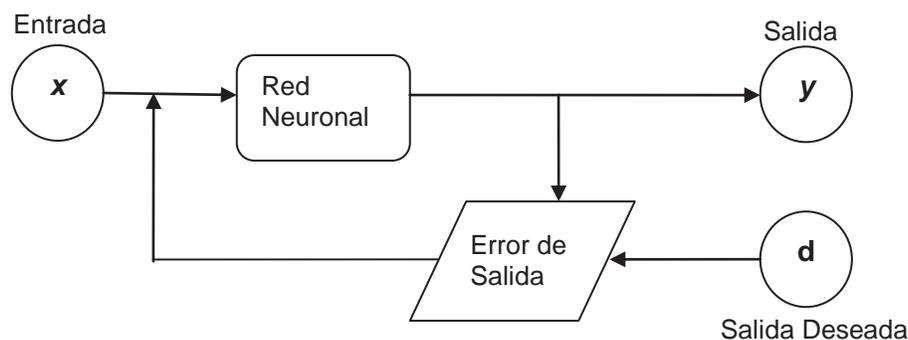


Figura 4.2. Aprendizaje Supervisado

En aprendizaje supervisado se utiliza el método de *Corrección del Error*, que consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error cometido en la salida.

4.2.2 Modelos de Aprendizaje Supervisado según Valor Deseado

En pronósticos se suele usar técnicas de suavizado de datos para reducir el efecto de los valores atípicos (outliers). La implementación de modelos como el de la Figura 3.4, en que tanto para la entrada como para el valor deseado se utilizan datos crudos, suele ser de difícil pronóstico dada la existencia de valores de comportamiento imprevisible.

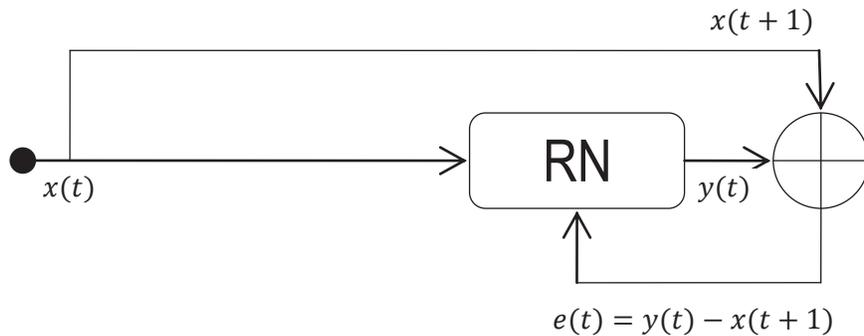


Figura 3.4. Entrada Cruda - Valor Deseado Crudo

Para reducir este efecto, suelen usarse técnicas de suavizado que se aplican a la señal, acercándola a la tendencia. Esto da lugar a modelos como el de la Figura 4.4, en que tanto la entrada como el valor deseado corresponden a data suavizada.

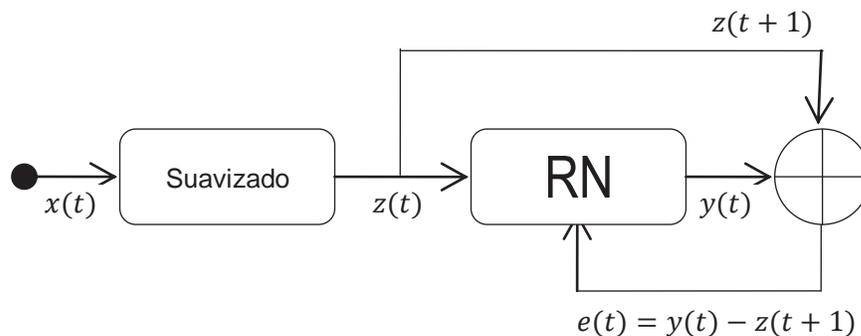


Figura 4.4. Entrada Suavizada - Valor Deseado Suavizado.

En la Figura 4.5 se presenta un modelo mixto, en que las entradas a la red son suavizadas y para el valor deseado se utilizan datos crudos.

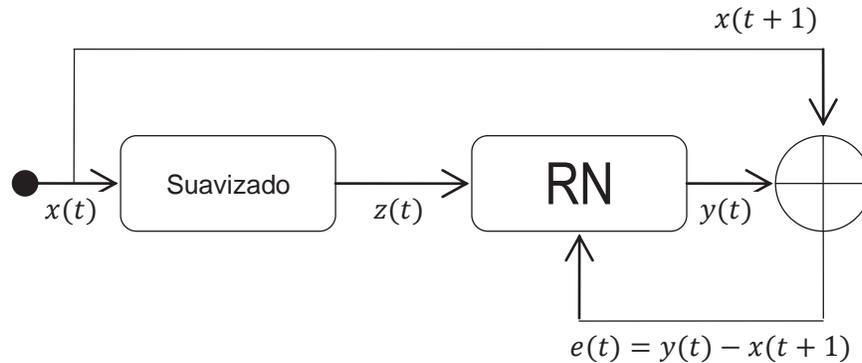


Figura 4.5. Entrada Suavizada - Valor Deseado Crudo.

4.2.3 Aprendizaje No Supervisado

En el aprendizaje no supervisado, a la red se le proporcionan únicamente los estímulos de entrada y ella ajusta los pesos de sus interconexiones utilizando solamente los estímulos y la salida calculada por la red. El aprendizaje no supervisado puede describirse como la estimación de la función de densidad de probabilidad $p(X)$ que describe la distribución de patrones X pertenecientes al espacio de entrada. El objetivo es descubrir patrones o características de los datos de entrada sin la ayuda de una fuente externa. Muchos algoritmos de aprendizaje no supervisado, básicamente, realizan agrupaciones (clustering) de los patrones de entrenamiento.

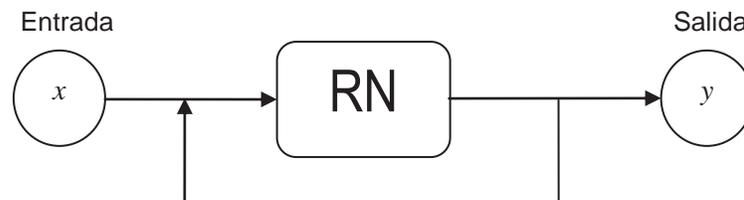


Figura 4.6: Aprendizaje No Supervisado

4.3 Tipos de estructuras de RNA

Según el atributo que se considere, existen diversas estructuras taxonómicas para clasificar la Redes Neuronales, tales como:

- Según la geometría: De capas o de racimos, bidimensionales o espaciales.
- Según el flujo de información: Hacia adelante (feedforward), o con retroalimentación (feedback).
- Según el número de neuronas: Mono, bi, tri o multineuronales.
- Según Topología: La Topología o Arquitectura es la organización de las neuronas en la red, formando capas o agrupaciones, más o menos alejadas de la entrada y salida de la red. Los parámetros fundamentales de la red son el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones ente neuronas. En términos topológicos se puede clasificar a las redes entre: redes de una sola capa y las redes con múltiples capas.

4.3.1 Redes Monocapa

En las redes monocapa, como la red HOPFIELD y la red BRAIN-STATE-IN-A-BOX, se establecen conexiones laterales entre las neuronas que pertenecen a la única capa de la red. Además pueden existir conexiones auto-recurrentes.

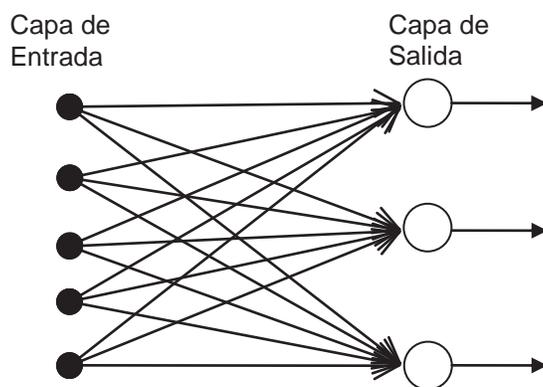


Figura 4.7. Red Neuronal Monocapa.

Las redes monocapa se utilizan típicamente en tareas relacionadas con lo que se conoce como auto asociación, por ejemplo para regenerar informaciones de entrada que se presentan distorsionadas o incompletas.

4.3.2 Redes Multicapa

Son aquellas donde las neuronas son agrupadas en varias capas. La forma para distinguir la capa a la que pertenece una neurona, es fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Normalmente, todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a la entrada de la red, y envían su señal de salida a una capa posterior, más cercana a la salida de la red. A estas conexiones se les denomina conexiones hacia delante o *feedforward*.

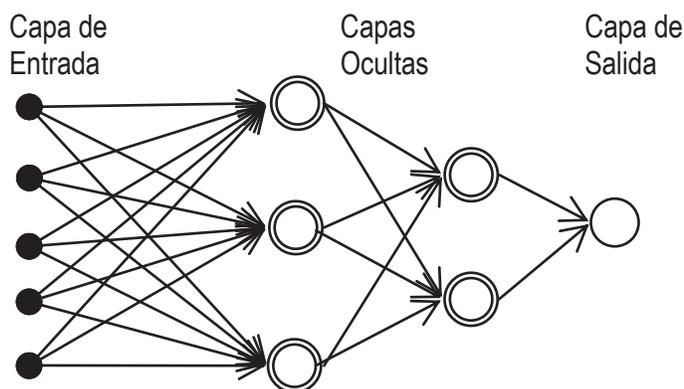


Figura 4.8: Red Neuronal Multicapa

En un gran número de estas redes también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores a las entradas de capas anteriores, A estas conexiones se les denomina conexiones hacia atrás o *feedback*.

Estas dos posibilidades permiten distinguir entre dos tipos de redes: con conexiones hacia adelante (*feedforward*), y redes que disponen de conexiones tanto hacia delante como hacia atrás (*feedforward/feedback*).

4.4 Redes Neuronales Feedforward y Recurrentes

4.4.1 Redes con Propagación hacia adelante (Feedforward)

Son redes donde las neuronas de una capa se interconectan con las neuronas de la siguiente capa, desde la capa de entrada hasta la capa de salida. Además el flujo de información en las intercapas se propaga en una sola dirección o hacia adelante (*feedforward*) a través de las capas de la red. No existen

conexiones hacia atrás, y normalmente tampoco auto recurrentes, ni laterales, excepto los modelos de red propuestos por Kohonen.

Se usan para transformar un conjunto de datos especificados en otros también especificados. El aprendizaje es supervisado ocurre a través de un proceso de ajuste de los pesos de las neuronas de la red. Son típicamente multicapa (ej. perceptrón multicapa). En éstas se distinguen una capa de entrada (*input layer*), una capa de salida (*output layer*), y algunas capas ocultas (*hidden layers*), de acuerdo a su posición dentro de la red (Figura 4.8). En estos modelos, el flujo de información es generalmente síncrono, es decir todos los estados de las neuronas se actualizan al mismo tiempo, o siguiendo alguna secuencia determinística. De este modo, las redes multi-capa son modelos estáticos como el modelo CMAC (“Cerebellar Model Articulation Control”) y las redes GMDH (“Group Method of Data Handling networks”).

En la

Figura 4.9 se muestra un ejemplo de este tipo de arquitectura.

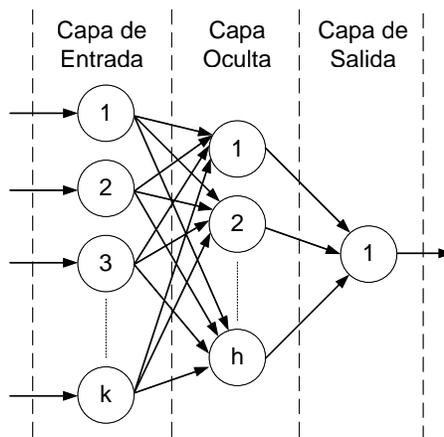


Figura 4.9. Red Neuronal Feedforward con una capa oculta y una salida.

Este tipo de red neuronal es el más utilizado en aplicaciones prácticas, obteniendo buenos resultados fundamentalmente como clasificadores de patrones y estimadores de funciones. Dentro de este grupo de redes neuronales, las más conocidas son: perceptron, adaline, madaline, linear adaptative memory, drive-reinforcement, backpropagation. Todas ellas son útiles en aplicaciones de reconocimiento o clasificación de patrones.

4.4.2 Redes Neuronales Recurrentes

En éste tipo de redes circula información tanto hacia adelante como hacia atrás durante el funcionamiento de la red. Para que eso sea posible existen conexiones *feedforward* y *feedback* entre las neuronas.

En general, suelen ser bicapas, existiendo por lo tanto dos conjuntos de pesos: los correspondientes a las conexiones *feedforward* de la primera capa (capa de entrada) hacia la segunda (capa de salida) y los de las conexiones *feedback* de la segunda a la primera. Los valores de los pesos de estos tipos de conexiones no tienen por qué coincidir, siendo diferentes en la mayor parte de los casos. La retroalimentación puede ocurrir entre neuronas de diferentes capas, de la misma capa e, incluso, entre una misma neurona.

Se caracterizan por su capacidad de modelar tanto la no linealidad como las componentes dinámicas de un sistema. Este tipo de estructura (bicapa) es particularmente adecuada para realizar una asociación de una información o patrón de entrada (en la primer capa) con otra información o patrón de salida en la segunda capa (lo cual se conoce como heteroasociación), aunque también pueden ser utilizadas para la clasificación de patrones. Son especialmente adecuadas para estudiar el comportamiento de sistemas no lineales.

También en este grupo de redes existen algunas conexiones laterales entre neuronas de la misma capa. Estas conexiones se diseñan como conexiones excitadoras (con peso positivo) o inhibitoras (con peso negativo), estableciéndose una competencia entre las neuronas correspondientes.

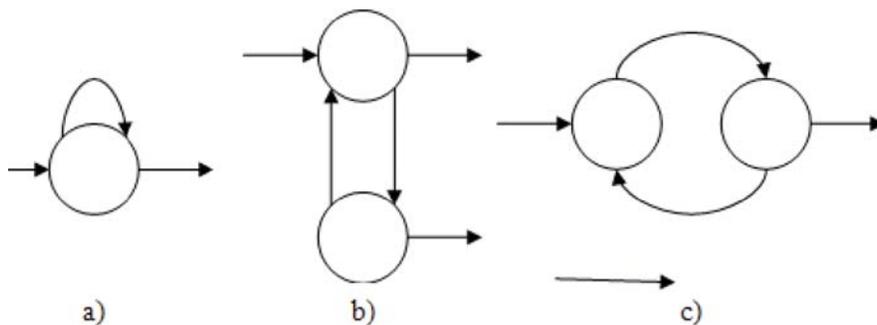


Figura 4.10. Tipos de Recurrencias: a) Conexión con la misma neurona, b) Conexión con neuronas de la misma capa y c) Conexión con neuronas posteriores y anteriores.

Similarmente al caso anterior, el flujo de información en este tipo de arquitecturas puede ser asíncrono, esto significa que los estados de cada neurona se actualizan de acuerdo al comportamiento interno de cada neurona. Así, la secuencia de actualización resulta estocástica.

Algunas redes tienen un funcionamiento basado en lo que se denomina resonancia, de tal forma que las informaciones en la primera y segundas capas interactúen entre sí hasta que alcanzan un estado estable. Esto permite un mejor acceso a las informaciones almacenadas en la red.

Los dos modelos de red de dos capas más conocidos son la red ART (*Adaptive Resonance Theory*) y la red BAM (*Bidirectional Associative Memory*).

Entre las clases de redes recurrentes más mencionadas en la literatura se encuentran las Redes de Jordan, de Elman, de Hopfield y de Kohonen,

El uso de este tipo de redes implica un aumento en el número de pesos o parámetros ajustables de la red permitiéndose, de esta manera, aumentar la capacidad de representación de la red dado que en las redes neuronales artificiales la información se representa de manera distribuida en los pesos de las conexiones y no en las neuronas. Por otro lado, la existencia de recurrencias en la red, junto con el aumento de los parámetros, por lo general, complica su proceso de aprendizaje.

En este caso, la activación ya no depende sólo de activaciones ocurridas en la capa anterior, sino también depende del estado o activación de cualquier otra neurona que se encuentre conectada a ella, o incluso su propia activación. Por lo tanto, es necesario incluir a la variable tiempo en la activación de la neurona:

$$a_i(t + 1) = f_i(\varphi_i(t)) \quad (6)$$

donde el índice j varía en el conjunto de todas las neuronas que se encuentran conectadas a la neurona i .

La presencia de $a_i(t + 1)$ en las activaciones de las neuronas recurrentes hace que éstas tengan un comportamiento dinámico o temporal, el que puede entenderse de dos formas:

- Evolución de las activaciones de la red hasta alcanzar un punto estable. Consiste en evolucionar la red (activaciones de las neuronas), desde un estado inicial dado por el patrón de entrada, hasta conseguir que todas estas activaciones de las neuronas no se modifiquen. En este momento se considera que la red ha alcanzado un estado estable, el cual representa el patrón de salida.
- Evolución de las activaciones de la red en modo continuo. En este caso, para cada instante de tiempo se tendrá una salida de la red, la cual depende de una entrada obtenida en el instante inmediatamente anterior. Este aprendizaje se puede llevar de dos maneras diferentes:

- i. Aprendizaje por épocas: La red evoluciona durante un intervalo de tiempo previamente definido, y cuando se alcanza este instante final se adaptan o modifican los pesos de la red. Una vez transcurrida una época, la red se reinicializa y se entra en una nueva época.
- ii. Aprendizaje en tiempo real o continuo: La ley de aprendizaje para modificar los pesos de la red se aplica en cada instante de tiempo, siempre y cuando exista la salida deseada para la red en dicho instante.

Dentro del grupo de redes recurrentes que presentan este modo de actuación, están las redes parcialmente recurrentes y totalmente recurrentes. Ambos tipos utilizan algoritmos de aprendizaje supervisados para la calibración de sus parámetros.

A pesar de la complejidad propia de las redes neuronales recurrentes y la dificultad, en ocasiones, para realizar su aprendizaje, han sido utilizadas para abordar distintos tipos de problemas. Por ejemplo, en reconocimiento de patrones y reconstrucción de procesos de escurrimiento de precipitaciones entre otros.

Existen distintos tipos de redes neuronales recurrentes, como las redes parcialmente recurrentes, ideadas por Jordan y Elman, con conexiones *feedforward*, a las que se le han añadido algunas conexiones hacia atrás.

4.4.2.1 Redes Completamente Recurrentes

Las arquitecturas completamente recurrentes corresponden a estructuras neuronales en las que se permite que cualquier neurona de la estructura presente conexiones hacia cualquier otra neurona. La Figura 4.11 presenta un diagrama de esta arquitectura. En ella se distinguen neuronas de entrada, $z(k)$, y neuronas de salida, $y(k)$. Así, si bien la arquitectura no presenta capas, es posible reordenar la estructura entre neuronas de entrada y salida y neuronas internas. La arquitectura reorganizada se presenta la Figura 4.12.

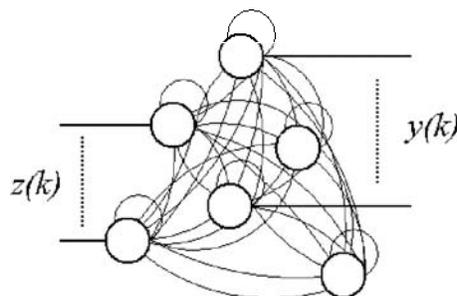


Figura 4.11. Red Completamente Recurrente.

Dentro de esta clasificación se encuentran las redes de Kohonen, las redes de Hopfield y sus variantes. Las redes de Hopfield han sido aplicadas en el modelamiento de sistemas dinámicos, así como variantes de dichas clases, aunque en general son empleadas en métodos de optimización, memorias asociativas, reconocimiento de patrones temporales y aplicaciones de inteligencia artificial. Por otra parte, las redes de Kohonen se emplean principalmente sobre problemas bidimensionales de reconocimiento de patrones.

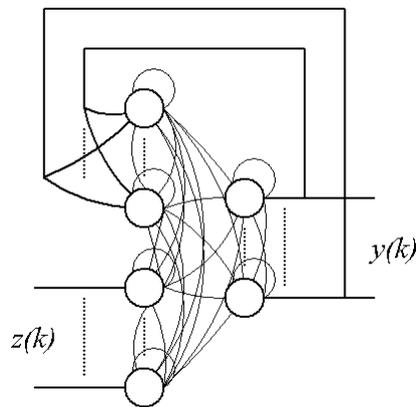


Figura 4.12: Red Completamente Recurrente Reorganizada en Capas.

4.4.2.2 Redes Parcialmente Recurrentes

Generalmente cuando se habla de redes neuronales parcialmente recurrentes, existe un grupo de neuronas especiales ubicadas en la capa de entrada, conocidas como **neuronas de contexto**, también llamadas neuronas de estado o neuronas auxiliares. De esta manera, se pueden distinguir 2 tipos de neurona en la entrada de la red: aquellas que actúan de entrada recibiendo las señales desde el exterior, y las neuronas de contexto (Figura 4.13) que actúan como neuronas receptoras de las conexiones recurrentes y funcionan como una memoria de la red en la que se almacenan las activaciones de las neuronas de una cierta capa de la red en el instante o iteración anterior (9).

Si se concatenan las activaciones de las neuronas de entrada y de las de contexto, una red parcialmente recurrente se puede ver como una red multicapa. De esta manera, en las redes parcialmente recurrentes, el cálculo de las activaciones de todas las neuronas se realiza como en una red multicapa sin recurrencias, es decir, desde la capa de entrada a la capa de salida, pasando por la capa oculta.

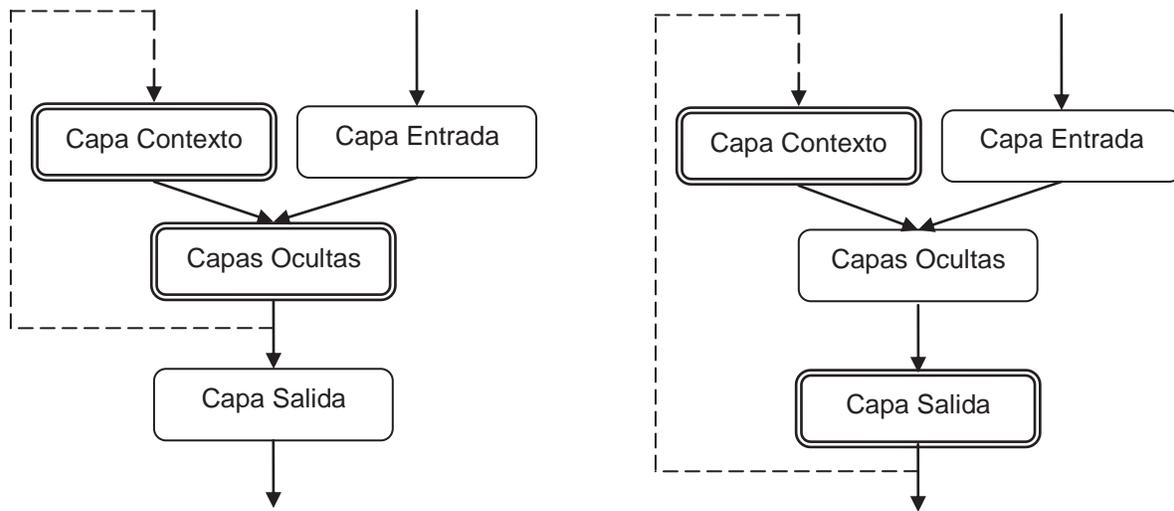


Figura 4.13. Esquemas de Redes Parcialmente Recurrentes.

Este tipo de arquitecturas se origina al emplear redes *feedforward* para el modelamiento de sistemas dinámicos. Las arquitecturas parcialmente recurrentes se construyen a partir de una topología tipo *feedforward*, sobre la que se consideran realimentaciones desde las capas de salida o desde las capas ocultas, hacia las capas de entrada. En esta categoría se incluyen las redes de Elman, las redes de Jordan y variantes de estas clases.

Jordan y Elman corresponde a las redes parcialmente recurrentes más conocidas. Su diferencia fundamentalmente radica en que en las redes de Jordan las neuronas de contexto reciben copias de las neuronas de la capa de salida y de sí mismas, y en las redes de Elman las neuronas de contexto reciben copias de las neuronas ocultas.

4.4.2.2.1 Red de Jordan

La primera arquitectura recurrente derivada a partir de una red *feedforward* fue planteada por Jordan en 1986. De acuerdo a la Figura 4.14, las redes de Jordan presentan una topología de capas similar a las redes *feedforward*, con la inclusión de una capa de contexto que realimenta las señales de salida hacia las capas ocultas (escondidas). En una red de Jordan existen retardos en todas las realimentaciones. El operador de retardo q^{-1} indica que la señal es retardada en una unidad de tiempo. Adicionalmente, se considera que los pesos de las realimentaciones desde la capa de salida hacia la capa de contexto son unitarios.

Jordan propuso una arquitectura donde la capa de entrada está dividida en dos partes: el conjunto de entradas externas y la realimentación de la activación de la capa de salida a través de conexiones de valor fijo.

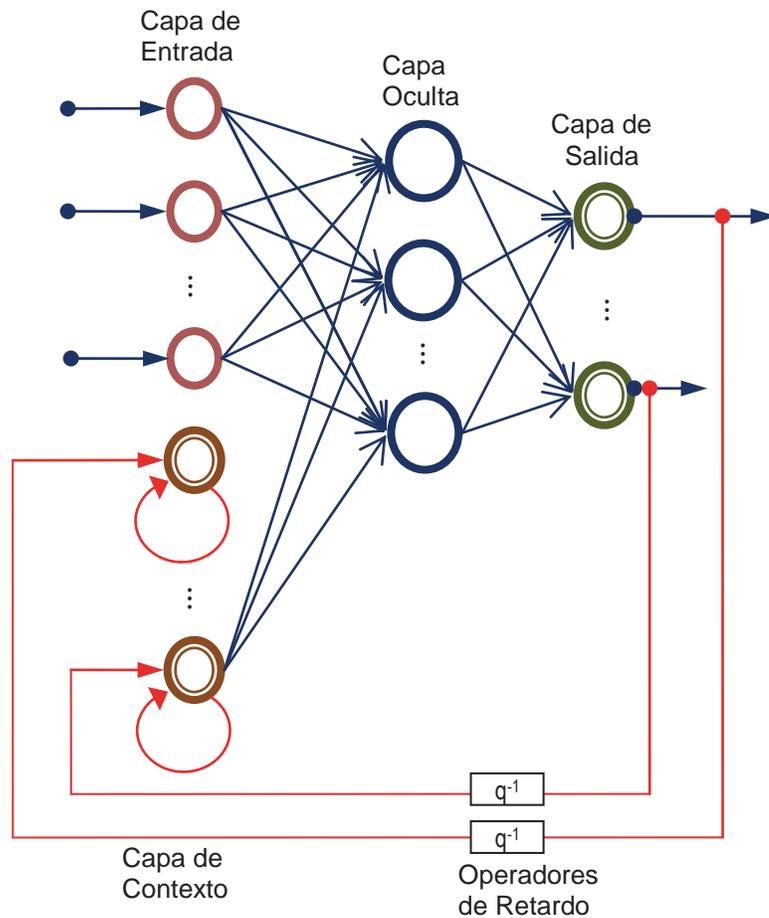


Figura 4.14. Red de Jordan.

Esta red se caracteriza porque las neuronas de contexto reciben una copia de las neuronas de salida de la red y de ellas mismas. Además, las conexiones recurrentes de la capa de salida a las neuronas de contexto llevan un parámetro asociado, que toma un valor constante positivo y menor que 1.

4.4.2.2.2 Red de Elman

En 1990, Elman modifica esta arquitectura, considerando realimentaciones desde capas ocultas hacia la capa de contexto, sin considerar realimentaciones locales. En la Figura 4.15 se muestra una red de Elman, también denominada red recurrente simple. El operador de retardo q^{-1} indica que la señal es retardada en una unidad de tiempo (10).

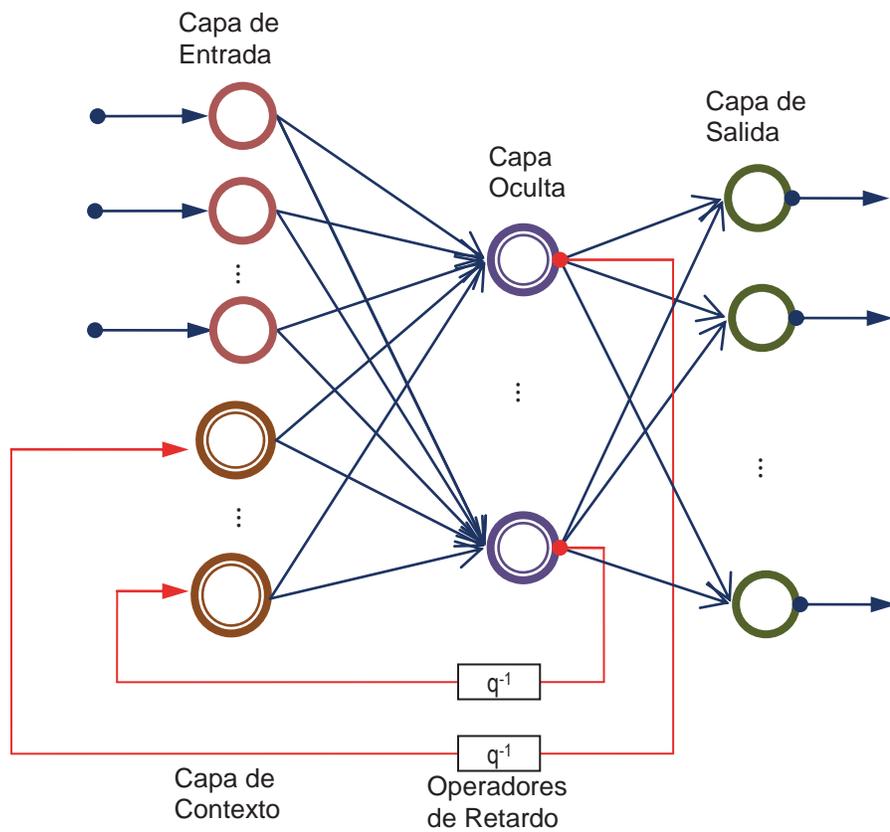


Figura 4.15. Red de Elman.

4.5 Ventajas de las Redes Neuronales Artificiales

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro (11). Por ejemplo, son capaces de aprender de la experiencia, generalizar de casos anteriores a nuevos casos, abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que presenten ventajas como:

- **Aprendizaje Adaptativo.** Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- **Auto-organización.** Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- **Tolerancia a fallos.** La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
- **Operación en tiempo real.** Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

- **Fácil inserción dentro de la tecnología existente.** Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

4.6 Algoritmos Evolutivos para Optimización (Aprendizaje) de Parámetros en RNA

4.6.1 Inteligencia de Enjambres

La Inteligencia de Enjambres (*Swarm Intelligence*, SI) se originó a partir del estudio de las colonias o enjambres de organismos sociales. Los estudios sobre el comportamiento social de los organismos (individuos) en enjambres, llevó al diseño de algoritmos de agrupamiento y optimización muy eficientes. Por ejemplo, estudios de simulación de la impredecible coreografía de bandadas de aves condujo al diseño del algoritmo de optimización de enjambre de partículas, y estudios sobre el comportamiento de las hormigas resultaron en algoritmos de optimización de colonias. Optimización por Enjambre de Partículas (PSO) es un enfoque de optimización estocástica, inspirado en el comportamiento social de las bandadas de aves. PSO es un procedimiento de búsqueda basado en la población donde los individuos, en forma de partículas, se agrupan en un enjambre. Cada partícula en el enjambre representa una solución probable al problema de optimización. En un sistema de obligaciones de servicio compartido, cada partícula "vuela" a través del espacio de búsqueda multidimensional, ajustando su posición en el espacio de búsqueda de acuerdo a su propia experiencia y la de las partículas vecinas. Una partícula por lo tanto hace uso de la mejor posición encontrado por sí mismo y la mejor posición de sus vecinos a posicionarse hacia una solución óptima. El efecto es que las partículas "vuelan" hacia un óptimo, mientras siguen buscando un área amplia alrededor de la mejor solución actual. El rendimiento de cada partícula (es decir, la "cercanía" de una partícula al mínimo global) se mide de acuerdo a una función de *fitness* predefinida que se relaciona con el problema que se resuelve. Las aplicaciones de PSO incluyen la aproximación de funciones, agrupamiento, la optimización de estructuras mecánicas, y los sistemas de resolución de ecuaciones. Los estudios de las colonias de hormigas han contribuido en abundancia para el conjunto de algoritmos inteligentes. El modelado de feromona depositada por las hormigas en su búsqueda de las rutas más cortas a las fuentes de alimentos, dieron como resultado el desarrollo de algoritmos de optimización de ruta más corta. Otras aplicaciones de la optimización de colonias de hormigas incluyen optimización de enrutamiento en redes de telecomunicaciones, la coloración de grafos, programación y solución del problema de asignación cuadrática. Los estudios sobre la construcción de nidos de hormigas y abejas dieron como resultado el desarrollo de algoritmos de optimización estructural y de agrupamiento.

4.6.2 Algoritmo Optimización por Enjambre de Partículas (PSO)

El algoritmo de Optimización mediante Enjambre de Partículas (*Particle Swarm Optimization*) o PSO, desarrollado en 1995 por el psicólogo-sociólogo James Kennedy y por el ingeniero electrónico Russell Eberhart, es un algoritmo bioinspirado, que se basa en una analogía con el comportamiento social de ciertas poblaciones de especies.

Se reconocen dos influencias básicas de inspiración en PSO:

- El movimiento de bandadas de aves o de bancos de peces, en las que cada individuo se desplaza mediante reglas simples de ajuste de su velocidad en función de las observaciones que realiza sobre los individuos próximos en la bandada.
- Un modelo social, en el que cada partícula representa una creencia, y las influencias entre individuos es la aproximación de unos individuos a otros por difusión de dichas creencias.

En líneas generales, el algoritmo busca encontrar el óptimo global de una función (función de ajuste, adecuación o *fitness*) mediante el movimiento de un conjunto de “partículas” (enjambre) en un espacio definido por el número de parámetros de la función. Cada partícula es una posible solución, es decir, un conjunto de parámetros, que se “evalúa” calculando el valor de *fitness* que les corresponde. En este movimiento, las partículas utilizan (12):

- Su conocimiento sobre el entorno (su adaptación, representado por su valor de *fitness*).
- Su conocimiento histórico o experiencias anteriores (su memoria, el resultado de su exploración pasada).
- El conocimiento histórico o experiencias anteriores de los individuos cercanos (memoria de vecindario).

Por otro lado, para completar la especificación del algoritmo, hay que definir algunos parámetros como criterio de parada, tipo de vecindario, valores de algunas constantes y forma de inicialización.

El enjambre de partículas (*swarm*) es un sistema multiagente, es decir, las partículas son agentes simples que se mueven por el espacio de búsqueda y que guardan (y posiblemente comunican) la mejor solución que han encontrado. Cada partícula tiene un *fitness*, una *posición* y un *vector velocidad* que dirige su movimiento. El movimiento de las partículas por el espacio está guiado por las partículas óptimas en el momento actual.

El enjambre comienza disperso por el espacio de búsqueda, pero con el tiempo las partículas convergen hacia una zona reducida del espacio en la que intensifican la búsqueda de la solución. Cada partícula se caracteriza por un vector de posición X_i , un vector de velocidad V_i , y una memoria que contiene el vector con la mejor posición anterior de la partícula, P_i . El vector de posición codifica una solución al problema de optimización, de modo que cada coordenada corresponde al valor de uno de los parámetros de la función que se desea optimizar.

4.6.2.1 Descripción del Algoritmo PSO

El algoritmo PSO consiste en un proceso iterativo y estocástico que opera sobre un enjambre de partículas, en que la posición de cada una representa una solución potencial al problema que se está resolviendo (Figura 4.17).

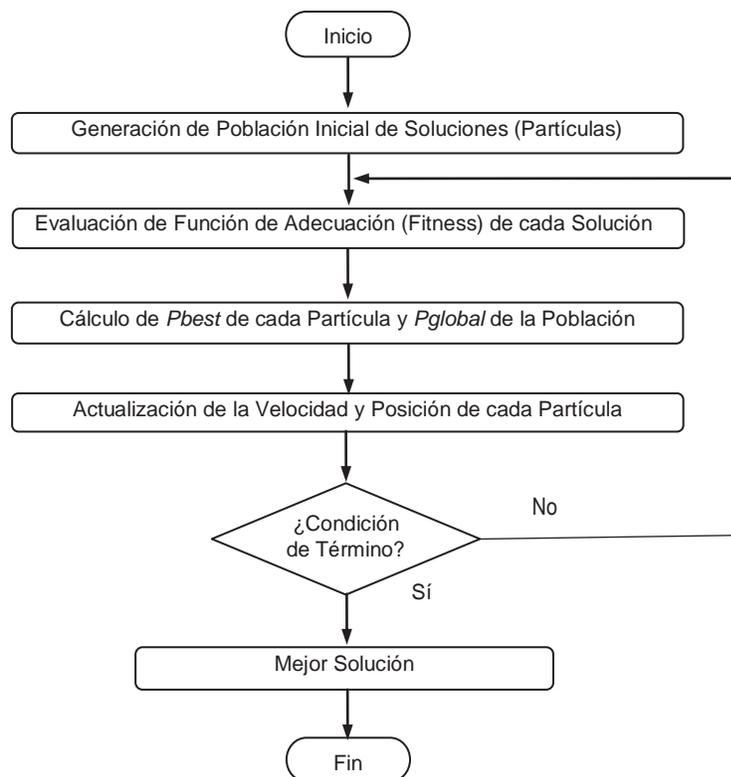


Figura 4.16: Flujo del Algoritmo PSO.

Generalmente, una partícula p_i está relacionada con tres vectores y dos valores de *fitness*:

- El vector $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]$ almacena la posición actual (localización) de la partícula en el espacio de búsqueda.
- El vector $pBest_i = [p_{i1}, p_{i2}, \dots, p_{in}]$ almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector de velocidad $V_i = [v_{i1}, v_{i2}, \dots, v_{in}]$ almacena el gradiente (dirección) según el cual se moverá la partícula.
- El valor de *fitness* $fitness_{X_i}$ almacena el valor de adecuación de la solución actual (vector X_{ij}).
- El valor de *fitness* $fitness_{pBest_i}$ almacena el valor de adecuación de la mejor solución local encontrada hasta el momento (vector $pBest_i$).

El enjambre se inicializa generando las posiciones y las velocidades iniciales de las partículas. Las posiciones se pueden generar aleatoriamente en el espacio de búsqueda (quizás con ayuda de un heurístico de construcción), de forma regular o con una combinación de ambas formas. Una vez generadas las posiciones, se calcula el *fitness* de cada una y se actualizan los valores de $fitness_{X_i}$ y $fitness_{pBest_i}$.

En cada iteración, la velocidad de cada partícula se actualiza utilizando la Ecuación (7). Dicha ecuación está escrita en forma escalar, de modo que, para cada vector considerado, se calcula la actualización en cada dimensión j de forma independiente. Puesto que los vectores varían en cada iteración, se utiliza el argumento t para indicar la iteración a la que se refiere una variable. Los términos c_1 y c_2 son parámetros fijos del algoritmo, y r_1 y r_2 son factores aleatorios entre 0 y 1, que se calculan de forma independiente en cada iteración y para cada dimensión. Por último, p_g corresponde con la mejor posición del mejor vecino de la partícula.

$$v_{ij}(t + 1) = v_{ij}(t) + c_1 r_1 (pBest_j - X_{ij}(t)) + c_2 r_2 (Pg_j - X_{ij}(t)) \quad \omega \sim U(0,1) \quad (7)$$

La Ecuación refleja la actualización del vector velocidad de cada partícula i en cada iteración.

Cuando la velocidad de todas las partículas se ha actualizado, esta se suma vectorialmente con la posición actual de la partícula para moverla a su posición para la siguiente iteración, según la Ecuación (8)

$$X_{ij}(t + 1) = X_{ij}(t) + v_{ij}(t + 1) \quad (8)$$

En la Ecuación (7) se suelen distinguir varios términos, que se suelen denominar haciendo referencia a la analogía social del algoritmo:

- Inercia ($v_{ij}(t)$), que representa la tendencia de un individuo a seguir explorando en la misma dirección.
- c_1 y c_2 : son ratios de aprendizaje (pesos) que controlan los componentes cognitivo y social.
- Cognitivo o individual, ($c_1\omega_1(PBest_{ij} - X_{ij})$), representa la distancia entre la posición actual y la mejor conocida por esa partícula, es decir, la decisión que tomará la partícula influenciada por su propia experiencia a lo largo de su vida, esto es la tendencia a permanecer próximo a posiciones (o creencias) que le dieron buen resultado en el pasado.
- Social, ($c_2\omega_2(P_{gj} - X_{ij})$), que representa la distancia entre la posición actual y la mejor posición del vecindario, es decir, la decisión que tomará la partícula según la influencia que el resto del enjambre ejerce sobre ella, esto es la imitación o seguimiento del mejor individuo conocido.

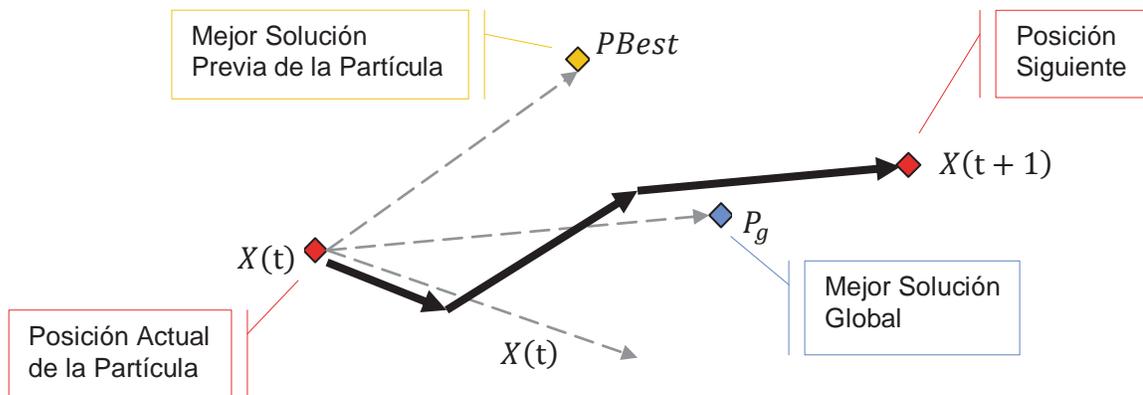


Figura 4.17: Esquema de Movimiento de la Partícula.

4.7 Indicadores de Bondad del Modelo

Se presenta una serie de métricas destinadas a evaluar la calidad del modelo.

4.7.1 Coeficiente de Correlación de Pearson (R)

El coeficiente de correlación de Pearson, R , es un índice que mide la relación lineal entre dos variables aleatorias cuantitativas. A diferencia de la covarianza, la correlación de Pearson es independiente de la escala de medida de las variables (19).

Sean y valores deseados e \hat{y} valores pronosticados, n = tamaño de la muestra en función de parejas, i en intervalo $[1,n]$, entonces:

$$R = \frac{n \sum y_i \hat{y}_i - \sum y_i \sum \hat{y}_i}{\sqrt{n \sum y_i^2 - (\sum y_i)^2} \sqrt{n \sum \hat{y}_i^2 - (\sum \hat{y}_i)^2}} \quad (9)$$

El valor de R varía en el intervalo $[-1,1]$:

Si $R = 1$, existe una correlación positiva perfecta. El índice indica una dependencia total entre las dos variables denominada relación directa: cuando una de ellas aumenta, la otra también lo hace en proporción constante (Figura 4.19 - Representación **a**).

Si $0 < R < 1$, existe una correlación positiva.

Si $R = -1$, existe una correlación negativa perfecta. El índice indica una dependencia total entre las dos variables llamada relación inversa: cuando una de ellas aumenta, la otra disminuye en proporción (Figura 4.19 - Representación **b**).

Si $-1 < R < 0$, existe una correlación negativa.

Si $R = 0$, no existe relación lineal. Pero esto no necesariamente implica que las variables son independientes, pueden existir todavía relaciones no lineales entre las dos variables (Figura 4.19 - Representación **c**).

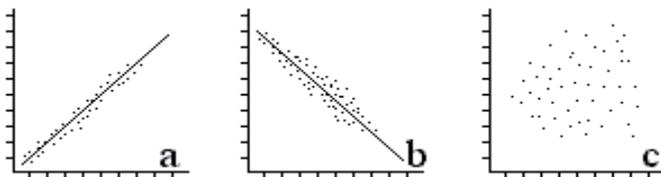


Figura 4.18: Representaciones de R : **a** Correlación Positiva Perfecta; **b** Correlación Negativa Perfecta; **c** No existe correlación.

4.7.2 Coeficiente de Determinación (R^2)

El Coeficiente de Determinación mide el porcentaje de variabilidad en Y que se puede explicar con el conocimiento de la variabilidad (diferencias) en la variable independiente X (19).

Para determinar el coeficiente de correlación entre dos variables y e \hat{y} , con n observaciones, se calcula:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \quad (10)$$

Donde y_i es el valor deseado e \hat{y}_i es el valor pronosticado por la regresión ajustada.

4.7.3 Error Cuadrático Medio (*Mean Squared Error, MSE*)

El error instantáneo del modelo se define (19) por:

$$e_i(t) = y_i(t) - \hat{y}_i(t) \quad (11)$$

Y el Error Cuadrático Medio (MSE) por:

$$MSE = \frac{1}{n} \sum_{i=1}^n e_i^2 \quad (12)$$

Esto es:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (13)$$

Donde n es el número de muestras, y_i valor observado (deseado) e \hat{y}_i valor estimado (pronosticado).

Representaciones aplicadas en adelante para cada métrica de calidad.

4.7.4 Raíz del Error Cuadrático Medio (*Root Mean Squared Error, RMSE*)

$$RMSE = \sqrt{MSE} \quad (14)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (15)$$

4.7.5 Error Medio (Mean Error, ME)

$$ME = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i) \quad (16)$$

4.7.6 Error Absoluto Medio (Mean Absolute Error, MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (17)$$

4.7.7 Error Porcentual Absoluto (Absolute Percentage Error, APE)

$$APE = \frac{|y_i - \hat{y}_i|}{|y_i|} (100) \quad (18)$$

4.7.8 Error Porcentual Absoluto Medio (Mean Absolute Percentage Error, MAPE)

$$MAPE = \frac{1}{n} \sum \frac{|y_i - \hat{y}_i|}{|y_i|} (100) \quad (19)$$

Entrega una referencia respecto a cuán grandes son los errores de pronóstico al compararlos con los valores reales de la serie.

4.7.9 Media de las BIAS (MB)

$$MB = \frac{(\bar{y} - \bar{\hat{y}})^2}{MSE} \quad (20)$$

4.7.10 Coeficiente U de Theil (Theil's U, U)

$$U = \frac{\sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}}{\sqrt{\frac{1}{n} \sum y_i^2 + \frac{1}{n} \sum \hat{y}_i^2}} \quad (21)$$

Si U tiende a 0, buena precisión del modelo. Si U tiende a 1, modelo no confiable

4.8 Selección de Variables

El problema de selección de variables ha adquirido especial interés en diversas áreas de aplicación donde el número de variables para un problema de pronóstico es superior al millar y muchas de ellas son redundantes o irrelevantes; entre estas áreas se encuentran la clasificación de documentos de la red o el análisis de expresión genética. El paradigma de la selección de variables trata de cubrir tres aspectos: mejorar el error de predicción, proporcionar modelos más rápidos y eficientes, y facilitar la interpretación de los modelos por parte de expertos (29).

La selección de variables aborda el problema de la reducción de la dimensión eligiendo el subconjunto de variables originales más informativo para mejorar el almacenamiento y procesamiento de los datos. El interés de estos métodos es que no alteran la representación original de los atributos preservando la información de las variables.

El proceso de selección se basa en desechar variables irrelevantes y/o redundantes aplicando criterios de evaluación. Los algoritmos metodológicos propuestos se separan en tres técnicas según su manera de combinar la selección de variables con el modelo predictivo (30):

- Método de Filtro
- Método de Wrapper
- Método Embebido

4.8.1 Métodos de Selección de Variables

4.8.1.1 Método de Filtro

En este método la evaluación y selección de variables forma parte del preprocesado de los datos y es independiente del predictor (Figura 4.20). Suele ser preferible a otros métodos de selección debido a su escalabilidad, son sencillos, rápidos computacionalmente y se pueden evaluar con distintos modelos predictivos. Los métodos de filtro, además, son utilizados en ocasiones como primer paso de un selector en dos etapas, el cual realiza una primera selección de variables que es refinada después mediante un método de Wrapper.

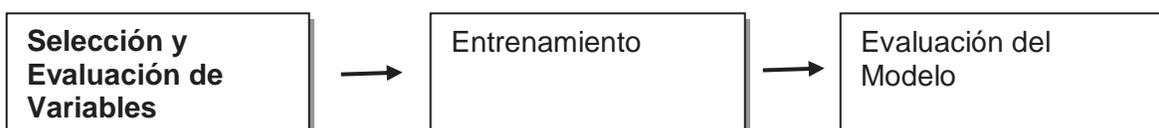


Figura 4.20. Método de Filtro.

4.8.1.2 Método de Wrapper

Este método es utilizado a modo de caja negra para otorgar puntuaciones a las variables en función del acierto (Figura 4.21). Para ello se realiza una doble búsqueda, primero en el espacio de variables donde se generan distintos subgrupos de atributos, que posteriormente son evaluados entrenando y validando el modelo predictivo. Para converger al conjunto óptimo (o subóptimo) de variables es necesario el empleo de algoritmos de búsqueda, ya sean aleatorios o heurísticos. Todo este proceso es más complejo y costoso que en los algoritmos de filtrado, aumenta el riesgo de sobreajuste y depende enormemente del modelo predictivo. Su ventaja es que puede explotar mejor las posibilidades de la predicción al aprovechar la interacción entre entrenamiento y validación y tienen en cuenta las dependencias entre variables de forma natural, cosa que muchos métodos de filtrado no. Posee estrategias de búsqueda, las más comunes son las búsquedas secuenciales, de selección hacia delante (Forward) o eliminación hacia atrás (Backward).

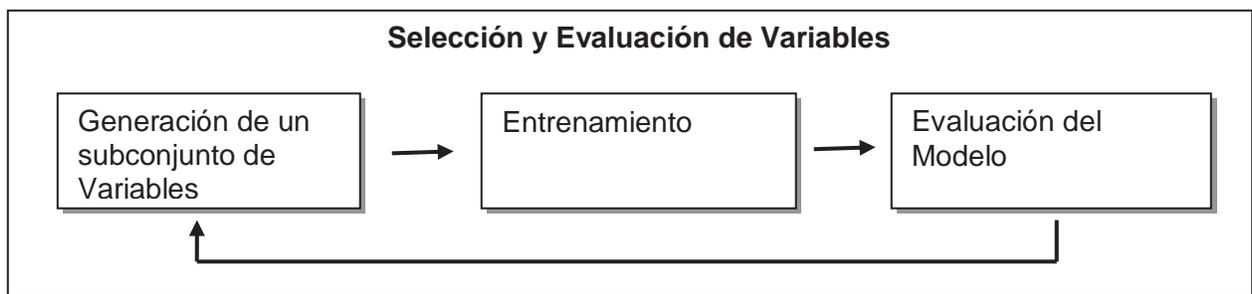


Figura 4.21. Método de Wrapper.

4.8.1.3 Método Embebido

En el método embebido la selección de las variables se realiza en el proceso de entrenamiento del modelo predictivo a partir del valor de una función objetivo a optimizar (Figura 4.22). Son menos estudiados y utilizados que los anteriores, sin embargo aprovechan mejor los datos y son más rápidos que los Wrapper al no necesitar evaluar cada subconjunto de variables.

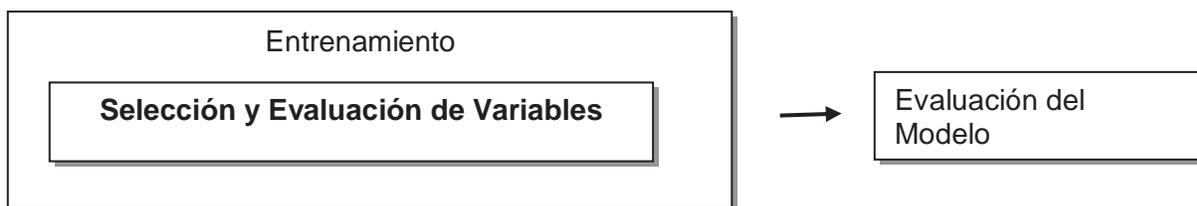


Figura 4.22. Método Embebido.

4.8.2 Métodos Algorítmicos de Selección de Variables explicativas en la Regresión

Uno de los aspectos más importantes en el proceso de selección es que de las variables candidatas a ser explicativas de la variable dependiente, se obtenga un subconjunto que resulte suficientemente informativo. En la práctica, la selección del variables explicativas de los modelos de regresión se deja en manos de procedimientos automáticos con criterios de calidad para evaluar la capacidad explicativa. Los procedimientos más usuales se clasifican en los siguientes métodos(27):

4.8.2.1 Método Algorítmico Backward:

Se comienza por incluir todas las variables disponibles y se van eliminando del modelo de una en una según su capacidad explicativa. En concreto, se va eliminando aquella que presente una menor evaluación con la variable dependiente, hasta obtener el subconjunto satisfactoriamente explicativo.

4.8.2.2 Método Algorítmico Forward:

Se comienza por un modelo que no contiene ninguna variable explicativa y se añade como primera de ellas a la que presente una mayor evaluación del criterio aplicado, con la variable dependiente. En los pasos sucesivos se va incorporando al modelo aquella variable que presenta mayor capacidad explicativa. El procedimiento se detiene cuando el incremento en la evaluación del predictor debido a la inclusión de una nueva variable explicativa en el modelo, ya no es importante.

4.8.2.3 Método Algorítmico Stepwise:

Es uno de los más empleados. En el primer paso se procede como el inicio del método Forward, pero a diferencia de éste en el que cuando una variable entra en el modelo ya no vuelve a salir, en el procedimiento Stepwise es posible que la inclusión de una nueva variable haga que otra que ya estaba en el modelo resulte redundante y sea “expulsada” de él.

4.8.3 Algoritmo de Selección de mínima Redundancia y Máxima Relevancia (mRMR)

Un ejemplo reconocido de selección por *Filtro*, tipo *Stepwise*, es el algoritmo “*mRMR*”, Mínima Redundancia-Máxima Relevancia, que selecciona la variable más relevante según un criterio de evaluación o calidad y va añadiendo iterativamente el atributo más relevante de los restantes y menos redundante con los ya seleccionados.

El algoritmo *mRMR* aporta valor dado que las variables relevantes de un proceso de predicción pueden tener información redundante. Un conjunto redundante de entradas candidatas no sólo aumenta la carga de cálculo de la fase de entrenamiento, sino también conduce a la degradación de la exactitud del pronóstico (32).

Como método de filtro, es independiente del modelo y con buen rendimiento y escalabilidad. También es un algoritmo que tiene en cuenta las relaciones entre variables. Genera el conjunto óptimo mediante una búsqueda incremental y utiliza generalmente como criterio de relevancia y redundancia, la *correlación* entre variables para evaluar relaciones lineales y la medición de *información mutua* para estimar relaciones no lineales entre ellas.

4.8.4 Criterio Coeficiente de Correlación de Pearson (R)

El Coeficiente de Correlación de Pearson (19) mide qué tanto se relacionan linealmente dos variables entre sí. La fuerza de la correlación se mueve entre -1 , cuando las variables tienen una relación negativa perfecta; y 1 , cuando tienen una relación positiva perfecta. Cuando R vale 0 , por tanto, no existe relación lineal entre las variables. La fórmula matemática de esta métrica se describe en punto 4.7.1 perteneciente a módulo *Indicadores de Bondad del Modelo, Estado del Arte*.

4.8.5 Criterio Información Mutua Normalizada (IMN)

La Información Mutua (*Amjady y Keynia, 2009*) (31), *IM*, es la medida más general de independencia estadística entre dos variables aleatorias, siendo capaz de cuantificar la información que estas comparten incluyendo las relaciones no lineales, punto al que no llega la correlación. La información mutua es, como la entropía, una medida de gran importancia dentro de la teoría de la información, define la información aportada por una variable aleatoria sobre otra.

Sean x e y dos variables aleatorias continuas y $p(x)$ y $p(y)$ sus funciones de densidad de probabilidad marginales, y $p(x,y)$ su función de densidad de probabilidad conjunta, se define su información mutua como (31):

$$IM(x, y) = \iint p(x, y) \log \frac{p(x,y)}{p(x)p(y)} dx dy \quad (22)$$

Como en la mayoría de los cálculos que incluyen funciones de densidad, el principal problema es estimar estas densidades a partir de los datos. Para variables discretas la ecuación se transforma en lo siguiente:

$$IM(x, y) = \sum \sum p(x, y) \log \frac{p(x,y)}{p(x)p(y)} \quad (23)$$

Es fácil comprobar que así definida, la información mutua es siempre positiva (las funciones de densidad son ≥ 0 en todo punto) y simétrica, es decir,

$$IM(x, y) = IM(y, x) \quad (24)$$

$$0 \leq IM(x, y) \leq IM \text{ Máxima } (y, x)$$

La información mutua será cero cuando las variables sean independientes y máxima cuando ambas tengan misma distribución binomial.

Para ajustar los valores de la IM a una escala de rango $[0,1]$, dado que posee un valor máximo de IM propio, se aplica normalización. Una opción de este proceso es(31):

Sea x una serie de N valores de Información Mutua, entonces

$$IM \text{ Normalizada} = x \text{ Normalizado} = \sqrt{1 - e^{-2x}} \quad (25)$$

Quedando x Normalizado en el intervalo $[0,1]$.

5 MARCO TEÓRICO DEL TRABAJO

El presente trabajo plantea generar y evaluar modelos de pronóstico basados en *redes neuronales parcialmente recurrentes de Jordan (RNRJ)*, con un modelo de aprendizaje *supervisado* con *entrada suavizada* con *promedio móvil desfase 3*, y con *valor deseado suavizado*.

Como función de transferencia se plantea *las unidades de suma (Transferencia Aditiva)*. Para la activación neuronal se experimentará con las funciones de *Identidad, Sigmoidal* y *Gaussiana*. Los algoritmos de aprendizaje serán *PSO (Optimización por Enjambre de partículas)* y variantes PSO con *Peso de Inercia Dinámico (IPSO)* y PSO con *Adaptación Dinámica (DAPSO)* (12), utilizando como función de costo el *Error Cuadrático Medio (ECM)*.

Para la selección de variables se aplicará Método de Filtro con algoritmo *mRMR (mínima Redundancia-Máxima Relevancia)* (32), con criterios *Correlación de Pearson (R)* e *Información Mutua Normalizada (NMI)* (31).

Para medir la bondad del modelo se usará como criterio principal el *Coefficiente de Determinación (R^2)* sin perjuicio de la aplicación de otras métricas de calidad.

5.1 Red Neuronal Parcialmente Recurrente de Jordan (RNRJ)

El tipo de red con que se trabajará es la recurrente de Jordan.

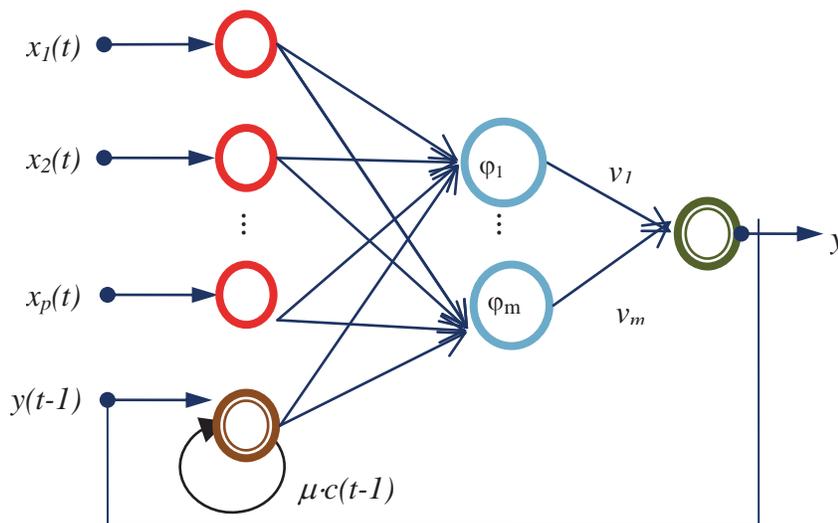


Figura 19 Red Parcialmente Recurrente de Jordan con una salida, sin capa de retardo.

En una red recurrente de Jordan, sin capa de retardo (Figura 19), la activación de cada neurona de contexto en el instante de tiempo o iteración t , denotada por $c_i(t)$, viene dada por:

$$c_i(t) = \mu \cdot c_i(t-1) + y_i(t-1) \quad (23)$$

donde $i = 1, 2, \dots, m$; μ es la influencia (retención) de pasado $\in [0, 1]$ y m corresponde al número de salidas de la red.

Además:

$$y(t-1) = (y_1(t-1), y_2(t-1), \dots, y_m(t-1)) \quad (24)$$

corresponde al vector de salida de la red en el instante $t-1$.

El resto de las activaciones se calculan como en una red multicapa con conexiones hacia delante, la diferencia radica en el vector de entrada en el tiempo t , dado que corresponderá a la concatenación de las activaciones de las neuronas de entrada junto con las de contexto. Este vector $u(t)$ se define como sigue:

$$u(t) = (x_1(t), x_2(t), \dots, x_n(t), c_1(t), c_2(t), \dots, c_m(t)) \quad (25)$$

$x_i(t)$, corresponde a la señal que la red recibe desde el exterior.

Las neuronas de contexto tienen una función de activación lineal, como se visualiza en la ecuación (23). Esto hace que la activación de las neuronas de contexto se pueda desarrollar en el tiempo del siguiente modo:

$$c_i(t) = \mu^2 \cdot c_i(t-2) + \mu \cdot y_i(t-2) + y_i(t-1) \quad (26)$$

$$= \mu^{t-2} \cdot y_i(1) + \mu^{t-3} \cdot y_i(2) + \dots + \mu \cdot y_i(t-2) + y_i(t-1) \quad (27)$$

Y se obtiene entonces la siguiente expresión:

$$c_i(t) = \sum_{j=1}^{t-1} \mu^{j-1} \cdot y_i(t-j) \quad (28)$$

Para $i=1, 2, \dots, m$

En esta expresión se observa que las neuronas de contexto acumulan las salidas de la red en todos los instantes anteriores de tiempo, y el parámetro μ determina la sensibilidad de estas neuronas de retener dicha información. Así, valores cercanos a 1 permitirán memorizar estados muy lejanos al tiempo actual, y a medida que este valor tiende a 0, dichos estados tienen una menor presencia en la activación actual de las neuronas de contexto.

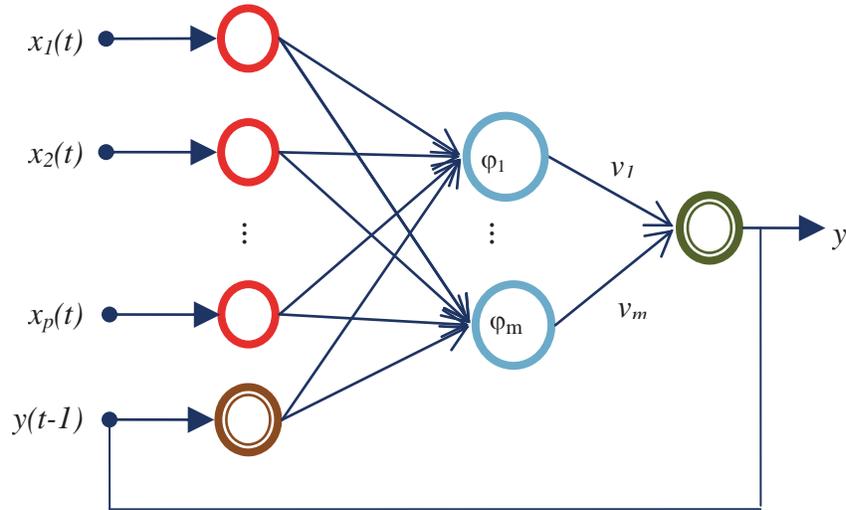


Figura 20 Red de Jordan con una salida, sin capa de retardo y sin retención de pasado.

En una red de Jordan, sin capa de retardo y sin retención de pasado (Figura 19), la activación de cada neurona de contexto en el instante de tiempo o iteración t , denotada por $c_i(t)$, viene dada por:

$$c_i(t) = y_i(t - 1) \quad (29)$$

donde $i = 1, 2, \dots, m$; $m = \text{número de salidas de la red}$.

Para el presente trabajo, en que el modelo requiere una salida, se tiene:

$$c(t) = y(t - 1) \quad (30)$$

5.2 Función de Transferencia

Se trabajará con función de transferencia aditiva. En el caso de las *Unidades de Agregación* (*Summation Units*), la señal de entrada a una Neurona Artificial se calcula como la suma ponderada de todas sus señales de entrada con sus pesos sinápticos correspondientes:

$$\varphi_j = \sum_i^n w_{ij} \cdot x_i + \theta \quad (31)$$

donde φ_j es la señal de entrada a la neurona j , w_{ij} los pesos sinápticos que llegan a la neurona j desde cada entrada x_i y n el número de entradas.

5.3 Funciones de Activación

Este estudio considera la utilización de las siguientes funciones de activación:

Función	Fórmula	Rango
Identidad	$f(x) = x$	$[-\infty, \infty]$
Sigmoidal (Logística)	$f(x) = \frac{1}{1 + e^{-x}}$	$[0, 1]$
Gaussiana	$f(x) = e^{-x^2}$	$[0, 1]$

Tabla 5.1 Funciones de Activación a utilizar en el trabajo.

5.4 Aprendizaje Supervisado con Recurrencia de Jordan

El aprendizaje será supervisado con valores suavizados en entrada y deseado. En RNRJ se visualiza como:

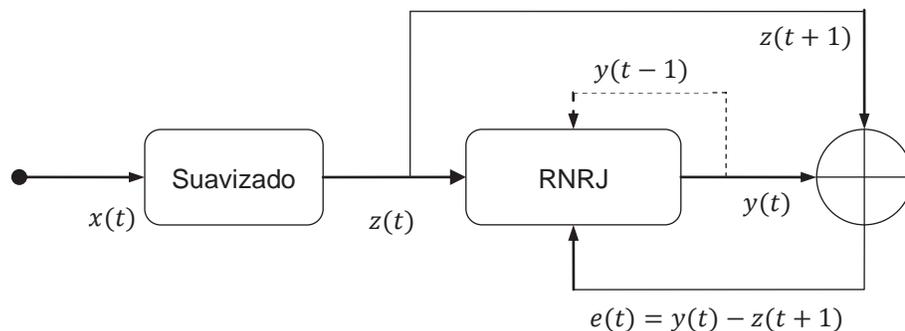


Figura 5.3 Red Recurrente con Entrada Suavizada - Valor Deseado Suavizado.

5.5 Suavizado con Promedio Móvil

La técnica del Promedio Móvil (*Moving Average*) es muy utilizada para obtener promedios sobre las señales a medida que éstas van ingresando. La idea del promedio consiste en tomar las N señales anteriores considerando la actual, sumarlas y dividir las por N . El promedio es "móvil" porque se recalcula cada vez que se obtiene una nueva señal tomando las N anteriores mencionadas.

Aplicado a una serie de tiempo, la señal suavizada y en el instante t se puede expresar como:

$$y(t) = \frac{1}{N} \sum_{i=0}^{N-1} x(t - i) \quad (32)$$

donde N es el desfase o ventana y x representa los valores *crudos* de la señal.

5.6 Algoritmo PSO

5.6.1 PSO Convencional

$$v_{ij}(t + 1) = v_{ij}(t) + c_1 r_1 (PBest_j - X_{ij}(t)) + c_2 r_2 (Pg_j - X_{ij}(t)) \quad \omega \sim U(0,1) \quad (32)$$

La Ecuación refleja la actualización del vector velocidad de cada partícula i en cada iteración. Cuando la velocidad de todas las partículas se ha actualizado, esta se suma vectorialmente con la posición actual de la partícula para moverla a su posición para la siguiente iteración, según la Ecuación (33):

$$X_{ij}(t + 1) = X_{ij}(t) + v_{ij}(t + 1) \quad (33)$$

5.6.2 PSO con Peso de Inercia Dinámico (IPSO)

En el algoritmo PSO estándar, la información sobre el mejor individual y el mejor global son compartidos por las próximas generaciones de partículas. Este algoritmo ofrece una mejora a través del uso de un peso de inercia dinámico que decrece en cada iteración.

El fundamento de PSO se basa en la hipótesis de que el intercambio social de información entre sus congéneres ofrece una ventaja evolutiva. Es así como el modelo estándar de PSO se cimienta en los siguientes dos factores: Primero, la memoria autobiográfica, que recuerda la mejor posición previa de cada partícula en el enjambre, segundo, el conocimiento global, que es la mejor solución previa de la población completa. Esto plantea que el intercambio de información entre los congéneres se logra mediante el empleo de la información disponible públicamente. No hay intercambio de información entre las

partículas, salvo la difusión de la mejor solución global. Por lo tanto, la población puede perder diversidad y se hace más probable que la búsqueda quede confinada en torno a mínimos locales si se ha convergido demasiado pronto en la búsqueda de los mejores globales encontrado hasta ahora.

En el algoritmo IPSO (17), cada partícula del enjambre comparte información mutua a nivel global y se beneficia de los descubrimientos y experiencias anteriores de todos los demás colegas durante el proceso de búsqueda.

$$v_{ij}(t+1) = \omega u^{-t} v_{ij}(t) + c_1 r_1 (PBest_j - X_{ij}(t)) + c_2 r_2 (Pg_j - X_{ij}(t)) \quad \omega \sim U(0,1) \quad (34)$$

$$X_{ij}(t+1) = X_{ij}(t) + v_{ij}(t+1) \quad (35)$$

donde ωu^{-t} es el peso de inercia, típicamente inicializado en el rango de $[0,1]$, $\omega \in [0,1]$, $u \in [1.0001, 1.005]$. Valores mayores de peso de inercia facilitan la exploración global y valores más pequeños tienden a facilitar la exploración local para el ajuste fino del área de búsqueda.

5.6.3 PSO con Adaptación Dinámica (DAPSO)

Esta variante de PSO propone un algoritmo con adaptación dinámica, usando una fórmula de actualización de velocidad de la partícula modificada, donde la aleatoriedad es reducida relativamente en el curso de actualización de velocidad y el peso de inercia de cada partícula es diferente. Por otra parte, este algoritmo presenta dos parámetros que describen el estado evolutivo del algoritmo: el factor de velocidad de evolución y el factor de grado de agregación (18).

$$v_i(t+1) = \omega_i(t) v_i(t) + c_1 r_1 (PBest_i - X_i(t)) + c_2 r_2 (Pg - X_i(t)) \quad \omega \sim U(0,1) \quad (36)$$

$$X_i(t+1) = X_i(t) + v_i(t+1) \quad (37)$$

Comparada con PSO convencional, la fórmula de actualización de velocidad tiene dos características diferentes:

- Los valores de $r1$ y $r2$ varían estocásticamente con el número de partículas e iteraciones. En otras palabras, en la iteración $(t+1)$ cada dimensión de la partícula i comparte los mismos valores aleatorios.
- El peso de inercia también varía con el número de partículas e iteraciones. Ya que el proceso de búsqueda de PSO es no lineal y altamente complicado, un peso de inercia decreciente linealmente puede no reflejar realmente el proceso de búsqueda. En esta variante del algoritmo, el peso de inercia se ve afectado por el estado evolutivo del algoritmo y determinado por el factor de velocidad de evolución de cada partícula y el factor de grado de agregación del enjambre.

El factor de velocidad de evolución y el factor grado de agregación

Esta variante del PSO define el factor de velocidad de evolución y el factor grado de agregación usando la siguiente fórmula:

(1) El factor de velocidad evolutiva

$$h_i(t) = \left| \frac{\min(F(PBest_i(t-1)), F(PBest_i(t)))}{\max(F(PBest_i(t-1)), F(PBest_i(t)))} \right| \quad (38)$$

donde $F(PBest_i(t))$ es el valor de *fitness* de $PBest_i(t)$. Bajo el supuesto y definición anterior, se obtiene que $0 < h_i \leq 1$. Este parámetro toma en cuenta la historia de funcionamiento de cada partícula, y refleja la velocidad de evolución de cada partícula, es decir, mientras menor sea el valor de h , mayor la velocidad.

(2) El grado de agregación

$$s = \left| \frac{\min(F_{best}(t), \bar{F}(t))}{\max(F_{best}(t), \bar{F}(t))} \right| \quad (39)$$

donde $\bar{F}(t)$ es el *fitness* promedio de todas las partículas del enjambre en la iteración t .

El factor de velocidad evolutiva y el grado de agregación son dos parámetros que caracterizan el curso de búsqueda de PSO, con los que debe variar el factor de inercia:

$$\omega_i(t) = \omega_{ini} - \alpha (1 - h_i(t)) + \beta s \quad (40)$$

donde ω_{ini} es el valor inicial de ω y que, en esta variante, puede asignársele 1. Ya que $0 < h_i \leq 1$ y $0 < s \leq 1$, se tiene que $1 - \alpha < \omega \leq 1 + \beta$. La elección de α y β está en el rango $[0,1]$.

5.6.4 Dimensión de la Partícula

Cada Partícula del enjambre representa una solución para los parámetros estimados, que en el caso de la red neuronal corresponden a los pesos sinápticos. Su composición y dimensión varía de acuerdo a los parámetros del modelo.

5.6.4.1 Dimensión de la Partícula RN Feedforward

Para una red *feedforward* con p entradas, con función de transferencia aditiva y activación sigmoideal, una capa oculta con h nodos y una salida, la dimensión θ de la partícula es:

$$\theta = p \cdot h + h \quad (41)$$

Con lo que la partícula se describe como:

$$P = [w_{11}, \dots, w_{1h}, w_{21}, \dots, w_{2h}, \dots, w_{p1}, \dots, w_{ph}, v_1, \dots, v_h] \quad (42)$$

donde W son los pesos de entrada y V los de salida.

5.6.4.2 Dimensión de la Partícula RNR de Jordan

Para una red parcialmente recurrente de Jordan con p entradas, con función de transferencia aditiva y activación sigmoideal, una capa oculta con h nodos y una salida, la dimensión θ de la partícula es:

$$\theta = (p + 1) \cdot h + h \quad (43)$$

Con lo que la partícula se describe como:

$$P = [w_{11}, \dots, w_{1h}, w_{21}, \dots, w_{2h}, \dots, w_{p1}, \dots, w_{ph}, w_{p+1,1}, \dots, w_{p+1,h}, v_1, \dots, v_h] \quad (44)$$

donde W son los pesos de entrada, incluyendo la entrada recurrente de Jordan en la posición $p+1$ y V los de salida.

5.7 Función de Costo

La función de Costo, también conocida como *Fitness*, *Ajuste*, *Adecuación* o *Adaptación*, se emplea para evaluar el nivel ajuste de cada solución del enjambre, *i-ésima* partícula *k-dimensional* $X_j = (a_{i1}, a_{i2}, \dots, a_{iK})$, se utiliza como función de costo el Error Cuadrático Medio (ECM o MSE, *mean square error*).

$$E(X_j) = \frac{1}{N} \sum_{i=1}^N (d_i - \hat{y}(u_i))^2 \quad (45)$$

donde N es el número de muestras del conjunto de entrenamiento. El valor deseado d_i y el vector regresor u_i se obtienen de:

$$d_i = [s(t)]$$
$$u_i = [s(t-1), s(t-2), \dots, s(t-p)]$$

donde p es el desfase autoregresivo.

5.8 Indicadores de Bondad del Modelo

El principal indicador de calidad de cada ejercicio de pronóstico corresponderá al **Coefficiente de Determinación (R^2)**, el cual mide el porcentaje de variabilidad entre los valores pronosticados y deseados. Para complementar la evaluación de bondad se analizará el costo por tiempo de proceso, que considera entrenamiento y validación, más los siguientes indicadores de calidad cuya formulación se encuentra descrita en el Estado del Arte, punto 4.7

- Coeficiente de Correlación (**R**)
- Error Cuadrático Medio (Mean Squared Error, **MSE**)
- Raíz del Error Cuadrático Medio (Root Mean Squared Error, **RMSE**)
- Error Medio (Mean Error, **ME**)
- Error Absoluto Medio (Mean Absolute Error, **MAE**)
- Error Porcentual Absoluto (Absolute Percentage Error, **APE**)
- Error Porcentual Absoluto Medio (Mean Absolute Percentage Error, **MAPE**)

5.9 Algoritmo de Selección de Variables

Por su independencia del modelo predictivo se decidió utilizar el **método de Filtro**, el cual se caracteriza por evaluar y seleccionar variables antes de iniciar el pronóstico (Figura 5.4).

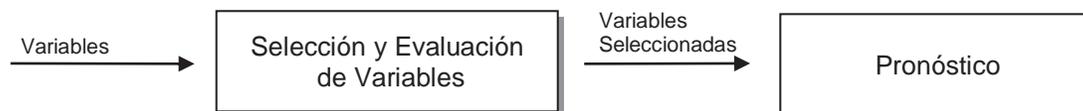


Figura 5.4 Proceso general Método de Filtro.

En el proceso de *Selección y Evaluación de Variables* se trabajará con algoritmo procedural **mRMR**, **m**ínima **R**edundancia-**M**áxima **R**elevancia (descrito posteriormente en punto 5.9.3), el cual propone utilizar un ciclo iterativo donde primero se rescatan las de mayor **relevancia** con variable dependiente y después, se descartan aquellas cuya información no aporta por estar contenida en otra u otras (variables **redundantes**).

Para evaluar la dependencia entre variables se aplican dos criterios de medición:

1. **Correlación de Pearson (R)** para estimar la relación lineal entre variables.
2. **Información Mutua Normalizada (IMN)** para observar el comportamiento no lineal entre variables.

Los resultados de estas mediciones conformarán la selección de variables al evaluar aquellas que superen valores mínimos establecidos (umbrales). Habrá un umbral para Relevancia y otro para Redundancia en cada criterio de medición. En selección por *Relevancia* las variables seleccionadas superarán un valor exigido de dependencia con variable interna (umbral “*Máxima Relevancia*”) y posteriormente, en descarte por *Redundancia*, se eliminará una variable de cada par de variables externas que sobrepase el tope aceptado de dependencia entre ellas (Umbral “*mínima Redundancia*”).

5.9.1 Criterio de Selección: Correlación de Pearson (R)

El coeficiente de Correlación de Pearson mide la relación lineal entre dos variables aleatorias cuantitativas, su formulación está descrita anteriormente en punto 4.7 del Estado del Arte, asociado a métricas de bondad del pronóstico.

El valor del índice de correlación varía en el intervalo $[-1,1]$. Si $R = 1$, existe una correlación positiva perfecta indicando una dependencia total dado que al aumentar una, la otra también aumenta en proporción. Si $R = -1$, existe una correlación negativa perfecta, es decir, cuando una de ellas aumenta, la otra disminuye en proporción. Si $R = 0$, no existe relación lineal.

De acuerdo a lo anterior, se aplicará valor absoluto de este indicador (valor real sin signo) para medir la dependencia lineal en el rango $[0,1]$.

5.9.2 Criterio de Selección: Información Mutua Normalizada (IMN)

5.9.2.1 Información Mutua (IM)

La información mutua, IM, mide la información que comparten dos variables aleatorias continuas, analizando su relación no lineal, estimación que no abarca la correlación(28).

Varía en el intervalo $[0, \text{Máxima IM}]$. Cuando las variables son independientes, conocer una no da información sobre la otra y viceversa, por lo tanto su información mutua es cero. En el otro extremo, si ambas variables son idénticas, conocer los valores de una determina conocer el valor de la otra y viceversa, y su IM tiene un valor máximo de medición.

De acuerdo a lo descrito en el Estado del Arte, punto 8.4.5, la formulación de la Información Mutua entre dos variables aleatorias continuas x e y , se expresa como:

$$IM(x, y) = \sum \sum p(x, y) \log \frac{p(x,y)}{p(x)p(y)} \quad (47)$$

Con $p(x, y)$ función de densidad de probabilidad conjunta y $p(x), p(y)$ funciones de densidad de probabilidad marginales.

Con las siguientes propiedades:

$$IM(x, y) = IM(y, x) \quad (48)$$

$$y \quad 0 \leq IM(x, y) \leq IM \text{ Maxima } (y, x)$$

La IM sera cero cuando las variables sean independientes y maxima cuando ambas tengan la misma distribucion (31).

5.9.2.2 Informacion Mutua Normalizada (IMN)

Para ajustar los valores de la IM a una escala de rango [0,1], dado que cada par de variables posee un valor maximo de IM propio y el algoritmo de seleccion evalua en base a umbrales genericos, se aplica la siguiente normalizacion (31):

Sea x una serie de N valores de Informacion Mutua, entonces

$$IM \text{ Normalizada} = x \text{ Normalizado} = \sqrt{1 - e^{-2x}} \quad (49)$$

Quedando x Normalizado en el intervalo [0,1].

5.9.3 Algoritmo mínima Redundancia-Máxima Relevancia (mRMR)

5.9.3.1 Determinación de Umbrales de Relevancia y Redundancia

El algoritmo mRMR trabaja con dos umbrales, uno para aplicar un primer filtro por Relevancia y otro para evaluar posteriormente la Redundancia. Una propuesta para determinar el valor de cada umbral, es generar una matriz de dependencia cruzada de variable interna y variables externas, de la cual por su simetría se considera el sector triangular superior (32). La tabla 5.2 muestra un ejemplo de dependencia de Correlación Lineal (R) cruzada de $x(t)$ y sus valores de 6 desfases ($t-1$ a $t-6$).

R	x_t	x_{t-1}	x_{t-2}	x_{t-3}	x_{t-4}	x_{t-5}	x_{t-6}
x_t	1	0.67	0.61	0.35	0.46	0.67	0.34
x_{t-1}	0.67	1	0.85	0.71	0.68	0.71	0.48
x_{t-2}	0.61	0.85	1	0.78	0.28	0.34	0.46
x_{t-3}	0.35	0.71	0.78	1	0.55	0.78	0.29
x_{t-4}	0.46	0.68	0.28	0.55	1	0.62	0.49
x_{t-5}	0.67	0.71	0.34	0.78	0.62	1	0.72
x_{t-6}	0.34	0.48	0.46	0.29	0.49	0.72	1

Tabla 5.2 Ejemplo de Matriz Cruzada de Correlación Lineal.

Con los valores de la primera fila de esta matriz, se analiza la Relevancia, tomando la dependencia de la variable dependiente x_t con sus independientes: x_{t-1} , x_{t-2} , x_{t-3} , x_{t-4} , x_{t-5} y x_{t-6} . El análisis se puede apoyar con una gráfica para determinar el umbral sobre el cual se visualizan aquellas variables más dependientes. En el ejemplo, la figura 5.5 visualiza un umbral de 0.5 de correlación que selecciona las variables más relevantes que corresponden a las variables de desfase 1 (0.67), 2 (0.61) y 5 (0.67).

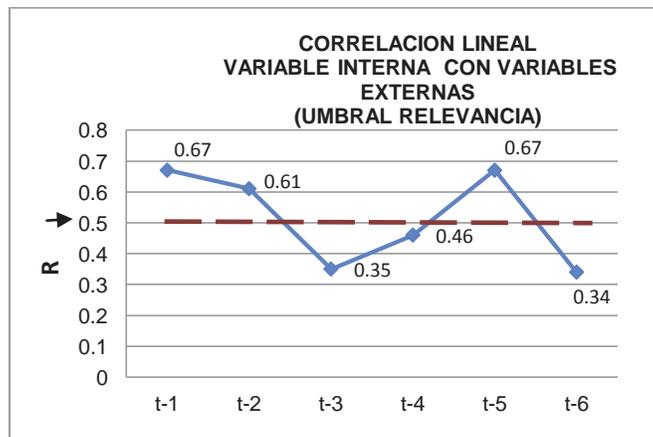


Figura 5.5 Gráfico de Ejemplo de Umbral de Relevancia.

Para determinar el umbral de Redundancia se analiza la dependencia entre las variables preseleccionadas por Relevancia, cuyo valor se indica en la matriz cruzada correlativa. Se procede de igual forma que para Relevancia, se grafican los valores y se aplica un análisis visual. La tabla 5.3 muestra los valores a analizar del ejemplo y la figura 5.6 muestra una gráfica de éstos a partir de la cual se puede estimar un umbral de 0.70 de correlación para considerar que sobre este valor existe redundancia.

R	x_t	x_{t-1}	x_{t-2}	x_{t-3}	x_{t-4}	x_{t-5}	x_{t-6}
x_t							
x_{t-1}			0.85			0.71	
x_{t-2}						0.34	
x_{t-3}							
x_{t-4}							
x_{t-5}							
x_{t-6}							

Tabla 5.3 Ejemplo de correlación de variables redundantes

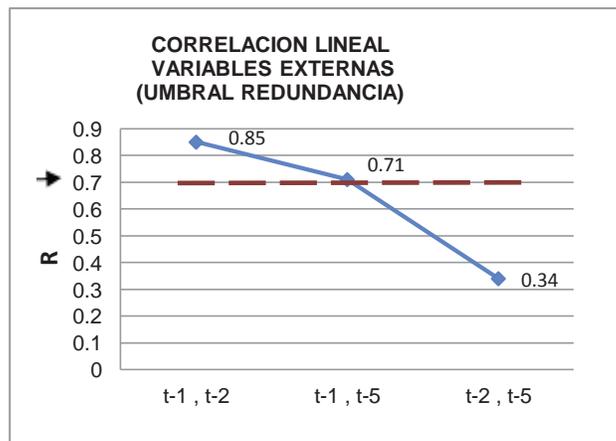


Figura 5.6 Gráfico de Ejemplo de Umbral de Redundancia.

La propuesta anterior presenta una metodología para determinar los umbrales de Relevancia y Redundancia sobre una mayor dimensión de variables que en ejemplo. Cuando esta dimensión sea muy elevada y el análisis visual no sea aplicable, una alternativa es generar la matriz cruzada y fijar como umbrales los valores de promedios, medianas u otra medida de tendencia estándar.

5.9.3.2 Desarrollo del Algoritmo mRMR

La figura 5.7 presenta el algoritmo mRMR con dos fases principales, la fase 1 (Relevancia) obtiene una preselección de variables filtrando aquellas que superan un mínimo de dependencia con la variable dependiente y la fase 2 (Redundancia) apunta a identificar aquellas redundantes para finalmente descartarlas del pronóstico.

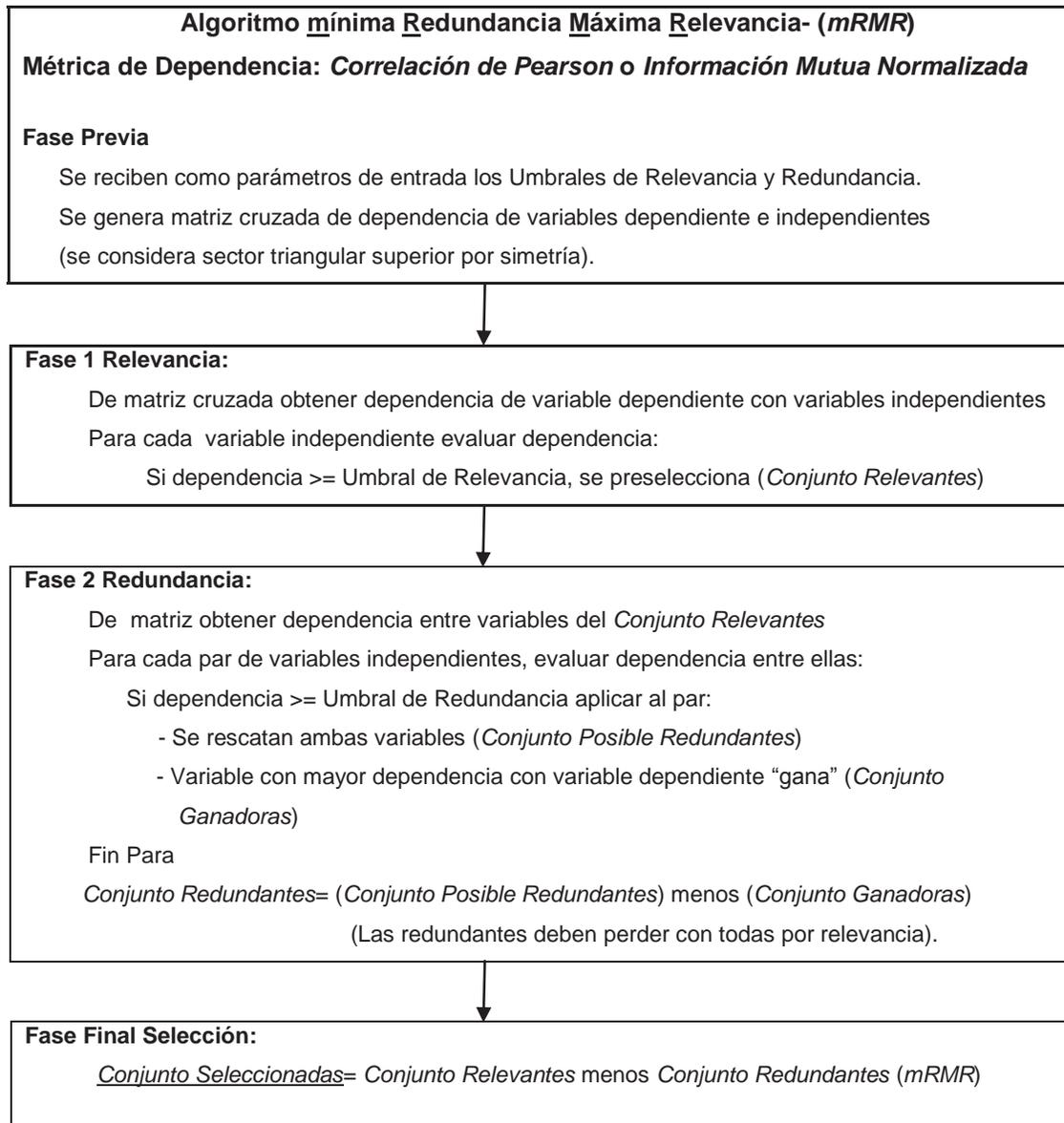


Figura 5.7 Algoritmo mRMR.

6 IMPLEMENTACIÓN

6.1 Preparación de datos

6.1.1 Datos fuente

Para la implementación del trabajo se dispone de datos históricos registrados por el Servicio Nacional de Pesca desde 1963 a 2006 (44 años), significando 528 muestras mensuales de cada variable ambiental a considerar (20) (Tabla 6.1).

VARIABLE	REPRESENTACIÓN	TIPO	MUESTRAS MENSUALES
Captura de Anchoveta	CA	Interna	528
Temperatura Superficial del Mar	TSM	Externa	528
Temperatura del mar de la Región Niño 3.4	N34	Externa	528

Tabla 6.1 Variables Ambientales del pronóstico

6.1.2 Normalizado Z-score

Dentro de las tareas de preparación de datos para optimizar su calidad, está la *transformación* a través de la *normalización*. La normalización estandariza los valores y los ajusta a una escala para caer en un rango pequeño y especificado. En la normalización *Z-score*, también llamada “*zero-mean normalization*”, los valores del vector V se normalizan con respecto a la media de V y a su desviación estándar transformando la distribución original en una distribución de media cero y variancia unitaria (23).

$$V \text{ normalizado } Z\text{-score} = (V - \bar{V}) / \sigma_V$$

La figura 6.1 visualiza la transformación por normalización Z-score de las variables CA, TSM y N34.

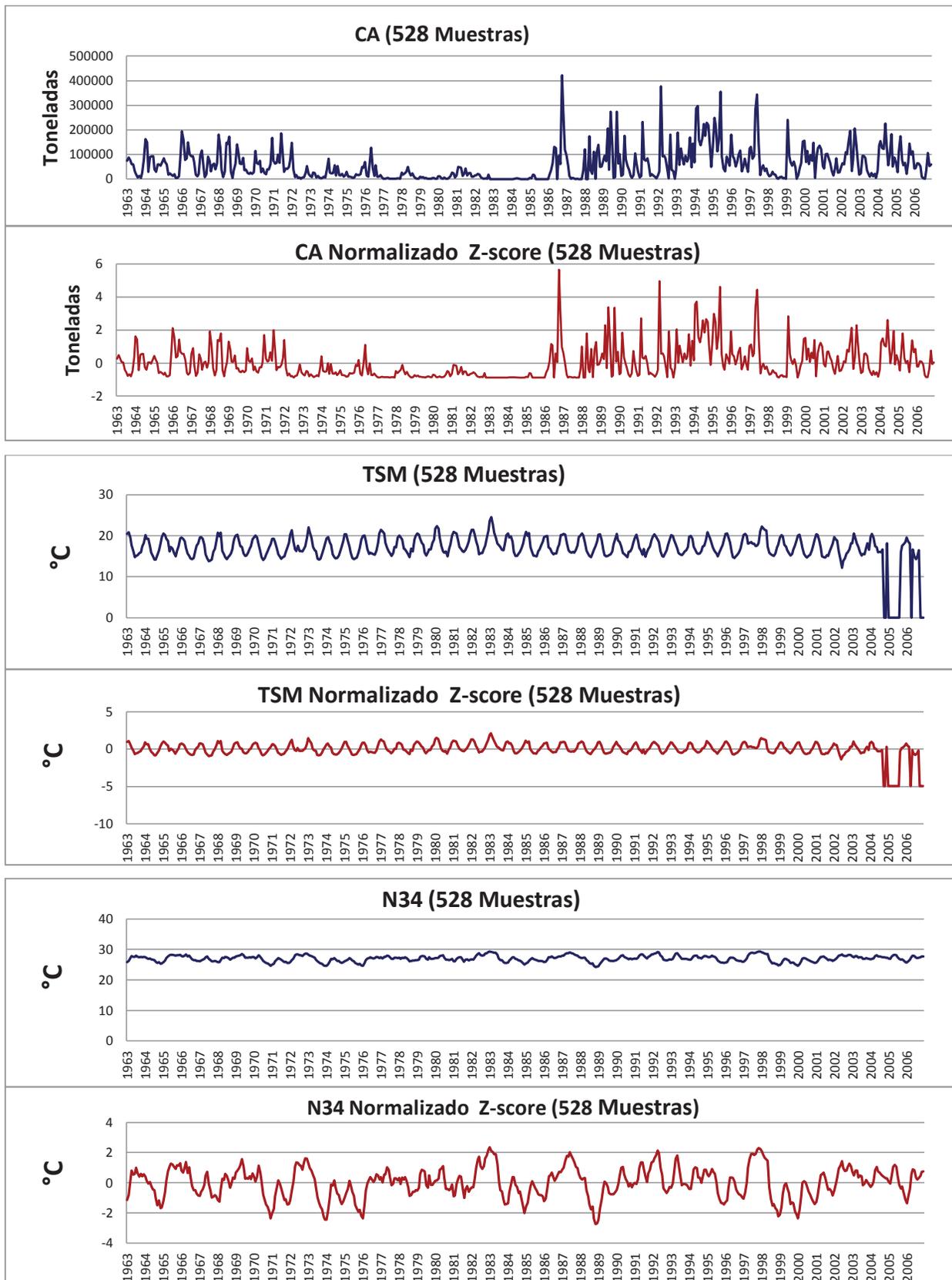


Figura 6.1 Normalización Z-score de datos fuente de variables CA, TSM y N34.

6.1.3 Suavizado con Promedio Móvil 3 (PM3)

Se aplica suavizado Promedio Móvil en los datos normalizados de cada variable, tomando 2 señales anteriores recientes y promediándolas con la señal actual. Para este caso, entonces, el factor del suavizado Promedio Móvil es 3, representándose por PM3. La figura 6.2 grafica el resultado del suavizado de las variables ambientales trabajadas.

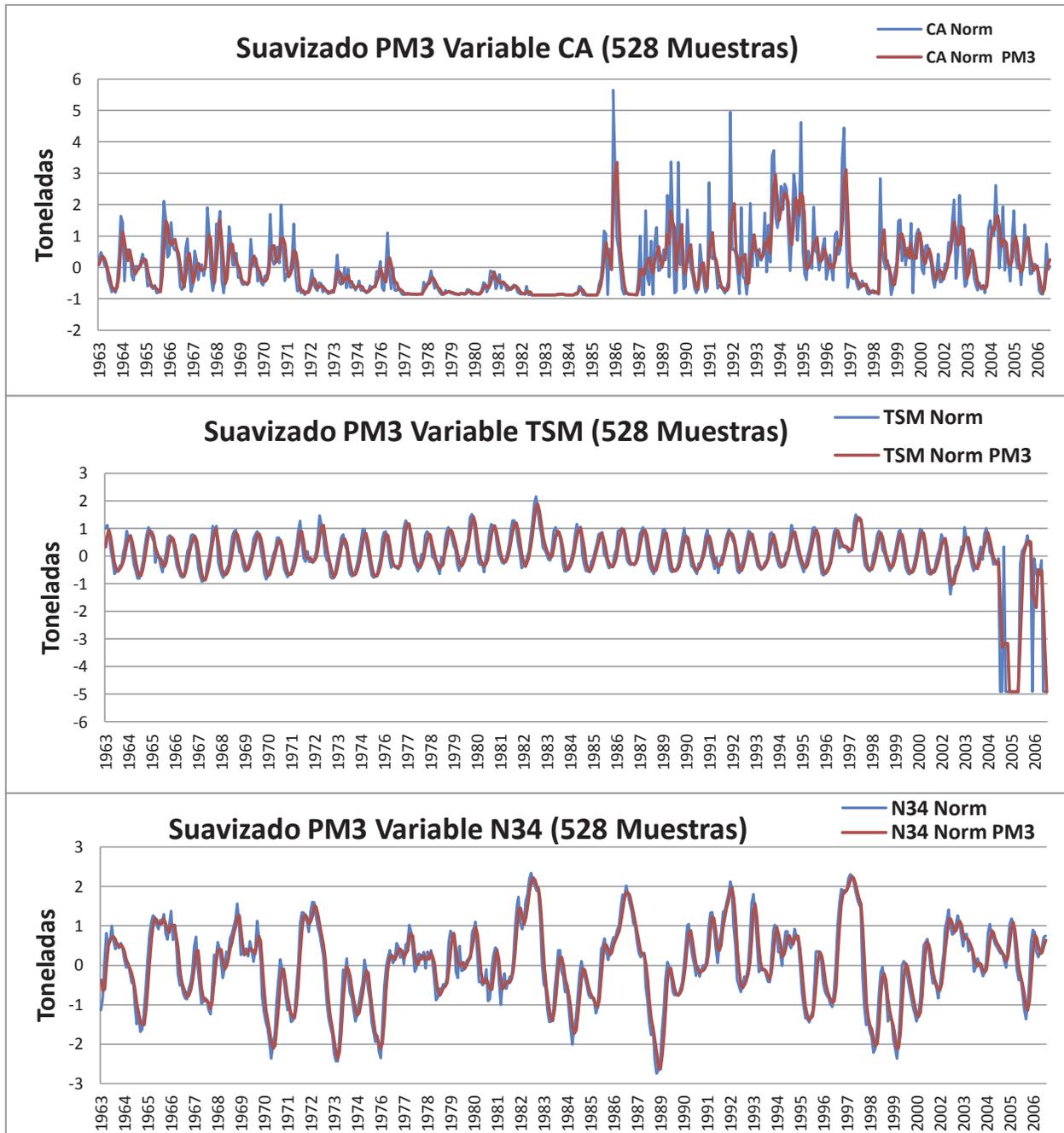


Figura 6.2 Suavizado PM3 de variables normalizadas CA, TSM y N34.

6.2 Experimentación

Se realiza una serie de 20 ejercicios de pronósticos para cada experimento. Un experimento considera un modelo neuronal de topología única, con un algoritmo de aprendizaje, con entrada univariada o multivariada, desfase y métrica de dependencia entre variable interna y externas en caso de multivariado (tabla 6.2).

La topología neuronal mencionada considera cantidad de entradas, nodos ocultos, partículas e iteraciones. Los algoritmos de aprendizaje evaluados son PSO y variantes IPSO y DAPSO. Las métricas para medir dependencia en selección de variables son Correlación de Pearson (R^2) e Información Mutua Normalizada (IMN).

TOPOLOGÍA RN	APRENDIZAJE	VARIABLES (EN DESFASES)	DESFASE DE REGRESIÓN	DEPENDENCIA (Selección Multivariado)
(Entradas, Nodos Ocultos, Salida)	PSO	CA	10	Correlación Pearson
Iteraciones	IPSO	TSM	20	IM Normalizada
Parámetros de algoritmo de aprendizaje	DAPSO	N34	30	
Función de Activación (Ident, Sigm, Gauss)			40	

Tabla 6.2 Experimento (parámetros de la experimentación).

Se inicia experimentando solución univariada con aprendizaje PSO, IPSO y DAPSO, se integran los resultados de todos los experimentos y se selecciona el mejor resultado evaluado con métrica de calidad Coeficiente de Determinación (R^2). La topología neuronal del mejor modelo de esta fase, con excepción de las entradas, será aplicada en cada experimento de solución multivariada (el criterio aplicado se basa en la mayor influencia de la variable CA con respecto a las variables externas).

6.3 Entrenamiento y Validación

En el proceso de calibración de un modelo neuronal debe existir validación. El término validación procura determinar si un modelo es correcto o no con respecto al sistema "real". La implementación requiere datos para entrenar y para validar. Después de evaluar proporciones de datos para las fases de entrenamiento y validación del pronóstico, se establece un **90%** de datos para entrenar y un **10%** para validar. El total de muestras se reduce por el valor del desfase de regresión (528 muestras menos períodos a desfasar).

La manera de automatizar los procesos de entrenamiento, validación es a través de un medio computacional que realiza los procesos iterativos de ajuste correspondiente (Software matemático *MATLAB*, abreviatura de *MATrix LABoratory*).

7 RESULTADOS EXPERIMENTALES

La bondad de las soluciones es evaluada con métricas de calidad descritas en el Estado del Arte (punto 4.7). Corresponden al Coeficiente de Determinación (R^2), Error Cuadrático Medio (MSE), Raíz del Error Cuadrático Medio ($RMSE$), Error Medio (ME), Error Absoluto Medio (MAE), Error Porcentual Absoluto (APE), Error Porcentual Absoluto Medio ($MAPE$) y Coeficiente U de Theil (tabla 7.1).

En base a estas estadísticas se puede seleccionar la topología neuronal del modelo que explique mejor la variabilidad del comportamiento del sistema. El Coeficiente de Determinación R^2 es la métrica que determina el mejor modelo, siendo aquel que posea el valor más cercano a 1 de esta medición. Los indicadores asociados a errores miden buen pronósticos cuando tienden a 0 y junto a la medición del tiempo de proceso, complementan la evaluación de calidad.

R^2	MSE	RMSE	ME	MAE	APE Peak	MAPE	U	Tiempo (seg.)
-------	-----	------	----	-----	-------------	------	---	------------------

Tabla 7.1 Métricas para evaluar el resultado experimental.

7.1 Resultado Experimental Solución Univariada

7.1.1 Parámetros de Experimentación

Parámetros de Experimentación del modelo RNRJ Univariado	
Entrenamiento - Validación	90% - 10%
Partículas	20, 30, 40, 50, 60
Iteraciones	400, 600, 800, 1000
Data Entrada - Valor Deseado	Suavizado PM3 - Suavizado PM3
Entradas (desfase regresivo CA)	6, 8, 10, 12 Variable Captura
Nodos Ocultos	6, 7, 8, 9, 10
Función Transferencia	Aditiva
Funciones Activación	Sigmoidal, Identidad, Gaussiana
Parámetros Aprendizaje PSO, IPSO, DAPSO	
Vel.Max., Vel.Min.	1.5, -1.5
C_1, C_2	1.5, -1.5
α_1, α_2	0.6, 1.0

Tabla 7.2 Parámetros Experimentación Solución Univariada.

7.1.2 Resultado Solución Univariada

La implementación univariada obtuvo como mejor topología **6 entradas, 7 nodos ocultos, y 40 partículas, con 600 iteraciones** y la función de activación **Sigmoidal**. El algoritmo de aprendizaje con mejor rendimiento fue **PSO**, acompañado de **DAPSO** como segundo mejor resultado (Tabla 7.3).

Se presenta el ranking ($R^2 \geq 0.84$) y comparación con proceso de Pseudoinversa. Éste último aplica regresión lineal con un cálculo inverso matemático sin el costo de aprendizaje del modelo neuronal.

Para la solución multivariada se deja estable la cantidad de **7 nodos ocultos, 40 partículas**, función de Activación **Sigmoidal**, además de los valores de los parámetros *Velocidades*, C_1 , C_2 , a_1 y a_2 de los algoritmos de aprendizaje PSO, IPSO y DAPSO.

R^2	Aprendizaje	Función Activación	Desfase de Regresión	Entradas CA (desfases)	Nodos Ocultos	Tiempo (seg)	Iteraciones	Partículas
0.84695	PSO	Sigmoidal	10	6	7	261.63	600	40
0.84293	DAPSO	Sigmoidal	10	6	7	270.77	600	40
0.84056	PSO	Sigmoidal	20	8	10	318.39	800	40
0.84030	PSO	Sigmoidal	20	8	10	302.45	800	40
0.84017	IPSO	Sigmoidal	10	6	7	266.78	600	40
0.83924	PSO	Sigmoidal	10	6	7	260.99	600	40

Tabla 7.3 Ranking Resultado Experimentación Solución Univariada con R^2 superior a 0.84 (inclusive).

Las evaluaciones de calidad que complementan la medición de R^2 se visualizan en Tabla 7.4.

Aprendizaje	R^2	R	MSE	RMSE	ME	MAE	APE Peaks	MAPE	U	Tiempo (seg.)
PSO	0.84695	0.92029886	0.001501	0.0387	0.0056	0.030294	17	26.056	0.10247	261.63
DAPSO	0.84493	0.91920074	0.001592	0.0399	0.0051	0.032055	16	29.252	0.10406	270.77
PSO	0.84456	0.91899946	0.001657	0.0407	0.0057	0.031686	18	30.309	0.10752	318.39
PSO	0.84030	0.91667879	0.001610	0.0401	0.0061	0.031700	18	30.808	0.10790	302.45
IPSO	0.84017	0.91660788	0.001612	0.0401	0.0063	0.032070	18	30.250	0.10906	266.78
PSO	0.83924	0.91610043	0.001592	0.0399	0.0059	0.032100	17	29.252	0.10996	260.99

Tabla 7.4 Indicadores de Bondad de los mejores resultados de Experimentación Solución Univariada.

7.1.3 Mejor resultado Solución Univariada

Se deduce de tabla 7.5 la mejor solución univariada.

R^2	APRENDIZAJE	ENTRADAS (desfases)	FUNCIÓN ACTIVACIÓN	NODOS OCULTOS	PARTICULAS	ITERACIONES
0.84695	PSO	CA (t-1 a t-6)	Sigmoidal	7	40	600

Tabla 7.5 Mejor Resultado Experimentación Solución Univariada.

7.1.4 Resultado Regresión Lineal con PseudoInversa

R^2	R	MSE	RMSE	ME	MAE	APE Peaks	MAPE	U	Tiempo (seg.)
0.82343	0.90743044	0.0014875	0.038568	0.10162	0.016396	19	23.97	0.17200	0.02926

Tabla 7.6 Indicadores de Bondad Solución con PseudoInversa.

Se observa un buen resultado de la regresión con Pseudoinversa ($R^2=0.82343$) considerando el costo ahorrado en tiempo (experimentación). El objetivo mantenido de obtener el modelo que explique mejor la variabilidad del sistema justifica el desarrollo del proceso neuronal ($R^2=0.84695$).

7.1.5 Gráfica de Correlación mejor Resultado Solución Univariada

Se presenta la gráfica comparativa, Figura 7.1, correspondiente a la representación de la correlación lineal R entre datos reales y pronosticados del mejor pronóstico $R^2= 0.84695$ del proceso de validación (10% de las muestras).

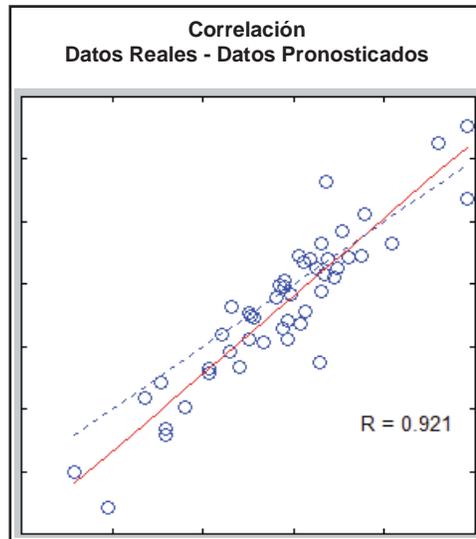


Figura 7.1 Correlación del mejor resultado univariado.

7.2 Resultado Experimental Solución Multivariada

7.2.1 Selección de Variables

7.2.1.1 Selección con algoritmo mRMR con criterio Correlación de Pearson

7.2.1.1.1 Análisis gráfico de dependencia con criterio Correlación de Pearson

Se analiza el comportamiento lineal (correlación) entre las variables a seleccionar en base a la gráfica de la Figura 7.2. Se determina un mínimo de 0.2 para el algoritmo mRMR.

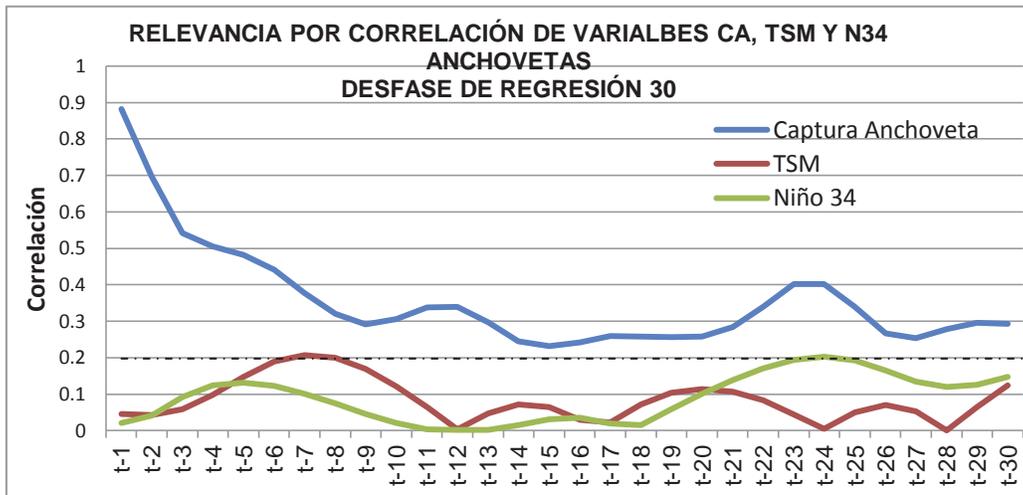


Figura 7.2 Gráfico de Relevancias de variables CA, TSM y N34 con Correlación de Pearson.

7.2.1.1.2 Resultado Algoritmo mRMR con Correlación de Pearson

Se visualiza en tabla 7.7 los parámetros del proceso algorítmico mRMR

Parámetros de Entrada a Experimentar mRMR Correlación de Pearson	
Umbral de Relevancia	0.2 (Resultado de análisis gráfico de relevancia)
Umbral de Redundancia	0.8 (Resultado de análisis gráfico de redundancia)
Desfase de la Regresión	06, 12, 18, 24, 30, 36, 42

Tabla 7.7 Parámetros Selección de Variables con criterio Correlación de Pearson.

Se logra seleccionar las variables CA, TSM y Niño 34 con desfase de regresión 30, Relevancia ≥ 0.2 y descarte de variables con Redundancia ≥ 0.8 .

Se visualiza en Tabla 7.8. lo expuesto gráficamente en Figura 7.3. La variable CA presenta alta relevancia en los desfases inmediatos, se mantiene resultado de solución univariada seleccionando hasta el desfase 6. La variable TSM en desfase 7 logra la relevancia 0.2 del umbral, y finalmente está el Niño34 que en un desfase más lejano, 24, alcanza la relevancia esperada.

DEFASE REGRESIÓN	UMBRAL RELEV	UMBRAL REDUN	VARIABLE	DEFASE	RELEVANCIA (sin redundancia)
30	0.2	0.8	CA	1	0.88
30	0.2	0.8	CA	2	0.7
30	0.2	0.8	CA	3	0.54
30	0.2	0.8	CA	4	0.5
30	0.2	0.8	CA	5	0.48
30	0.2	0.8	CA	6	0.44
30	0.2	0.8	CA	7	0.38
30	0.2	0.8	CA	8	0.32
30	0.2	0.8	CA	10	0.31
...
30	0.2	0.8	CA	26	0.27
30	0.2	0.8	CA	28	0.28
30	0.2	0.8	CA	29	0.3
30	0.2	0.8	TSM	7	0.21
30	0.2	0.8	N34	24	0.2

Tabla 7.8 Resultado Selección de Variables con mRMR, criterio Correlación de Pearson.

7.2.1.1.3 Resultado Selección de Variables con criterio Correlación de Pearson

La tabla 7.9 resume el resultado de la selección con criterio Correlación de Pearson.

CA						TSM	N34
t-1	t-2	t-3	t-4	t-5	t-6	t-7	t-24

Tabla 7.9 Variables seleccionadas con Correlación de Pearson.

7.2.1.2 Selección con algoritmo mRMR con criterio IM Normalizada

7.2.1.2.1 Análisis gráfico de dependencia con criterio IM Normalizada

Se analiza el comportamiento no lineal (Información Mutua) entre las variables a seleccionar en base a la gráfica de la Figura 7.3. Se determina un mínimo de 0.4 para el algoritmo mRMR.

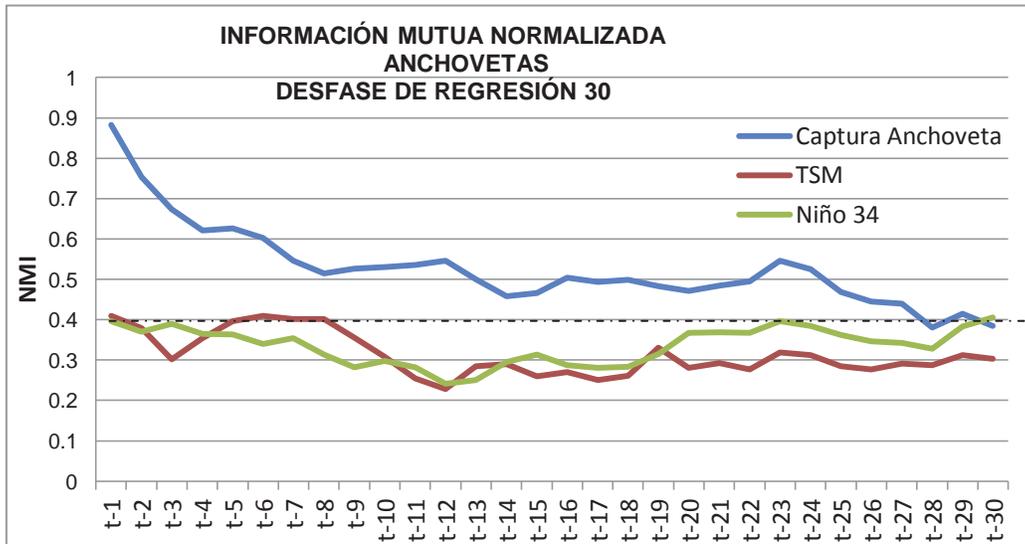


Figura 7.3 Gráfico de Relevancias de variables CA, TSM y N34 con IMN.

7.2.1.2.2 Resultado Algoritmo mRMR con IMN

Parámetros de Entrada a Experimentar mRMR IMN	
Umbral de Relevancia	0.4 (Resultado de análisis gráfico de relevancia)
Umbral de Redundancia	0.8 (Resultado de análisis gráfico de redundancia)
Desfase de la Regresión	06, 12, 18, 24, 30, 36, 42

Tabla 7.10 Parámetros Selección de Variables con criterio Correlación de Pearson.

La tabla 7.10 indica los parámetros del proceso mRMR con IMN. El resultado trabajó con las variables CA, TSM y Niño 34 con desfase de regresión 30, Relevancia ≥ 0.2 y descarte de variables con Redundancia ≥ 0.8 .

Se visualiza en Tabla 7.11 lo expuesto gráficamente en Figura 7.3. La variable CA presenta alta relevancia en los desfases inmediatos, se mantiene resultado de solución univariada seleccionando hasta el

desfase 6. La variable TSM en desfases 1 y 6 logra superar el umbral indicado y finalmente aparece Niño34 que con desfase más lejano, 30, alcanza el criterio exigido.

DESFASE REGRESIÓN	UMBRAL RELEV	UMBRAL REDUN	VARIABLE	DESFASE	RELEVANCIA (NMI)
30	0.4	0.8	CA	1	0.8830
30	0.4	0.8	CA	2	0.7542
30	0.4	0.8	CA	3	0.6735
30	0.4	0.8	CA	4	0.6445
30	0.4	0.8	CA	5	0.6258
30	0.4	0.8	CA	6	0.6020
30	0.4	0.8	CA	7	0.5457
30	0.4	0.8	CA	23	0.5457
30	0.4	0.8	CA	12	0.5455
...
30	0.4	0.8	CA	26	0.4449
30	0.4	0.8	CA	27	0.4399
30	0.4	0.8	CA	29	0.4149
30	0.4	0.8	TSM	6	0.4096
30	0.4	0.8	TSM	1	0.4089
30	0.4	0.8	N34	30	0.4055

Tabla 7.11 Resultado Selección de Variables con mRMR, criterio IMN.

7.2.1.2.3 Selección de Variables con criterio IMN

La tabla 7.12 resume el resultado de la selección con criterio IMN.

CA						TSM		N34
t-1	t-2	t-3	t-4	t-5	t-6	t-1	t-6	t-30

Tabla 7.12 Variables seleccionadas con IMN.

7.3 Resultados Modelo Multivariado

7.4 Resultado Experimental Solución Multivariada

7.4.1 Parámetros de Experimentación

La tabla 7.13 muestra los parámetros de la experimentación de la solución multivariada. Se considera para el modelo neuronal los parámetros del mejor resultado de la solución univariada : **7 Nodos Ocultos, 40 Partículas, función de activación Sigmoidal** y **parámetros** de los algoritmos PSO, IPSO y DAPSO (Tabla 7.3 Ranking Resultado Experimentación Solución Univariada).

Parámetros de Experimentación del modelo RNRJ Multivariado	
Validación	10%
Partículas	40
Iteraciones	600, 800, 1000
Data Entrada	Suavizado Promedio Móvil desfase 3
Valor Deseado	Suavizado Promedio Móvil desfase 3
Modelo Neuronal	Recurrente Jordan
Nodos Ocultos	7
Función Transferencia	Aditiva
Función de Activación	Sigmoidal
Aprendizaje PSO, IPSO y DAPSO	
Vel.Max., Vel.Min.	1.5 , -1.5
C ₁ , C ₂	1.5 , -1.5
α ₁	0.6
α ₂	1.0

Tabla 7.13 Parámetros Experimentación Solución Multivariada.

7.4.2 Resultado Solución Multivariada

El resultado de la solución multivariada se obtuvo de la experimentación con la topología neuronal y las variables que se indican en tabla 7.14, más los aprendizajes PSO, IPSO y DAPSO.

Función Activación	Desfase Regresión	Ent. Var.	Nodos Ocultos	Partículas	Criterio de Selección	Variables Seleccionadas (Entrada)		
Sigmoidal	10	6	7	40	(Univariado)	CA (t-1 a t-6)		
Sigmoidal	30	8	7	40	Correlación	CA (t-1 a t-6)	TSM(t-7)	N34(t-24)
Sigmoidal	30	7	7	40	Correlación	CA (t-1 a t-6)	TSM(t-7)	
Sigmoidal	30	7	7	40	Correlación	CA (t-1 a t-6)		N34(t-24)
Sigmoidal	30	9	7	40	IMN	CA (t-1 a t-6)	TSM(t-1,t-6)	N34(t-30)
Sigmoidal	30	8	7	40	IMN	CA (t-1 a t-6)	TSM(t-1,t-6)	
Sigmoidal	30	7	7	40	IMN	CA (t-1 a t-6)		N34(t-30)

Tabla 7.14 Experimentación Solución Multivariada.

El modelo neuronal que presentó mejor resultado logró un R^2 de **0.8566 (0.86)** con aprendizaje PSO y variables de entrada CA, TSM y N34 seleccionas por criterio de Correlación Lineal de Pearson (CA (t-1 a t-6), TSM(t-7) y N34(t-24)) . Este resultado supera la solución univariada de $R^2 = 0.8470$.

La tabla 7.15 visualiza los resultados de experimentos que superaron un R^2 de 0.84 integrando las soluciones univariada y multivariada. Se observa en ella que la variante de aprendizaje DAPSO se ubica en segundo lugar logrando un rendimiento muy cercano a su original PSO empleando las mismas variables de entrada. También se observa que los dos resultados con entradas seleccionadas con criterio IMN, experimentos 5 y 9, coinciden con los resultados con entradas seleccionadas por correlación de Pearson.

Exp.	R^2	Aprendizaje	Criterio de Selección	Variables			Iteraciones
1	0.8566	PSO	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
2	0.8553	DAPSO	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
3	0.8519	DAPSO	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
4	0.8490	IPSO	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
5	0.8490	IPSO	IMN	CA (t-1 a t-6)	TSM (t-1,t-6)	N34 (t-30)	1000
6	0.8470	PSO	(Univariado)	CA (t-1 a t-6)			600
7	0.8464	PSO	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
8	0.8450	PSO	Correlación	CA (t-1 a t-6)	TSM(t-7)	N34 (t-24)	1000
9	0.8450	PSO	IMN	CA (t-1 a t-6)	TSM (t-1,t-6)	N34 (t-30)	1000
10	0.8449	DAPSO	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
11	0.8447	ISPO	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
12	0.8429	DAPSO	(Univariado)	CA (t-1 a t-6)			600

Tabla 7.15 Ranking Resultado Experimentación Solución Integrada con R^2 superior a 0.84.

Las evaluaciones de calidad que complementan la medición de R^2 de los mejores seis resultados (hasta solución univariada, experimento 6, $R^2=0.84695$) se indican en Tabla 7.16.

Exp	Aprendizaje	R^2	R	MSE	RMSE	ME	MAE	APE Peaks	MAPE	U	Tiempo (seg.)
1	PSO	0.8566	0.9255	0.001409	0.0375	0.0055	0.030174	17	24.056	0.11247	421.46
2	DAPSO	0.8553	0.9248	0.001412	0.0376	0.0053	0.031051	18	26.252	0.11406	432.26
3	DAPSO	0.8519	0.9230	0.001657	0.0407	0.0050	0.031677	18	26.909	0.10752	439.96
4	IPSO	0.8490	0.9214	0.001610	0.0401	0.0061	0.032070	18	25.808	0.10890	446.47
5	IPSO	0.8490	0.9214	0.001612	0.0401	0.0063	0.032120	18	25.750	0.10906	468.42
6	PSO	0.8470	0.92030	0.001501	0.0387	0.0056	0.030294	17	26.056	0.10247	261.63

Tabla 7.16 Indicadores de Bondad de los mejores resultados de Experimentación Solución Multivariada.

7.4.3 Mejor resultado Solución Multivariada

Se deduce de tabla 7.17 la mejor solución multivariada con entrada ($CA_{t-1} a t-6$, TSM_{t-7} , $N34_{t-24}$).

R^2	APREND.	ENTRADAS (Seleccionadas por Correlación)			FUNCIÓN ACTIVACIÓN	NODOS OCULTOS	PARTIC.	ITERA.
0.8566	PSO	CA (t-1 a t-6)	TSM(t-7)	N34(t-24)	Sigmoidal	7	40	1000

Tabla 7.17 Mejor Resultado Experimentación Solución Multivariada.

7.4.4 Comparativo de Resultados Soluciones Univariada y Multivariada

La solución multivariada resulta con el mejor pronóstico del proceso general de experimentación, con una ganancia de R^2 de 1.12071 con respecto a la solución univariada (Tabla 7.18).

R^2	Ganancia R^2 (Dif / Mejor * 100)	APREND.	ENTRADAS (Univariada y Multivariada)			FUNCIÓN ACTIVACIÓN	NODOS OCULTOS	PARTIC.	ITERA.
0.8566	(Mejor)	PSO	CA (t-1 a t-6)	TSM(t-7)	N34(t-24)	Sigmoidal	7	40	1000
0.8470	1.12071	PSO	CA (t-1 a t-6)			Sigmoidal	7	40	600

Tabla 7.18 Comparativo Resultados Solución Univariado y Multivariado.

7.4.5 Comparativo de Resultados de Aprendizajes PSO, IPSO, DAPSO

La tabla 7.19 presenta los indicadores de bondad de los mejores resultados de los aprendizajes PSO, IPSO y DAPSO. La tabla 7.20 indica la Ganancia R^2 del mejor aprendizaje PSO (Diferencia R^2 / Mejor $R^2 * 100$) con respecto a variantes IPSO y DAPSO.

Aprendizaje	R^2	R	MSE	RMSE	ME	MAE	APE Peaks	MAPE	U	Tiempo (seg.)
PSO	0.8566	0.9255	0.001409	0.0375	0.0055	0.030174	17	24.056	0.11247	421.46
DAPSO	0.8553	0.9248	0.001412	0.0376	0.0053	0.031051	18	26.252	0.11406	432.26
IPSO	0.8490	0.9214	0.001610	0.0401	0.0061	0.032070	18	25.808	0.10890	446.47

Tabla 7.19 Comparativo Resultados de Aprendizajes PSO, IPSO y DAPSO.

Aprendizaje	R^2	Ganancia R^2 de PSO (Dif / Mejor * 100)	Criterio de Selección	Variables Seleccionadas			Itera.
PSO	0.8566	(Mejor)	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
DAPSO	0.8553	0.151763	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000
IPSO	0.8490	0.887229	Correlación	CA (t-1 a t-6)	TSM (t-7)	N34 (t-24)	1000

Tabla 7.20 Ganancia mejor aprendizaje PSO con respecto a variantes IPSO y DAPSO.

7.4.6 Gráficas del Mejor Resultado Solución Multivariada

Se presenta la gráfica comparativa, Figura 7.4, entre datos reales y pronosticados del mejor pronóstico $R^2 = 0.8566$ del proceso de validación (10% de las muestras) junto con la representación de la correlación lineal R de ambas series de datos (Figura 7.5):

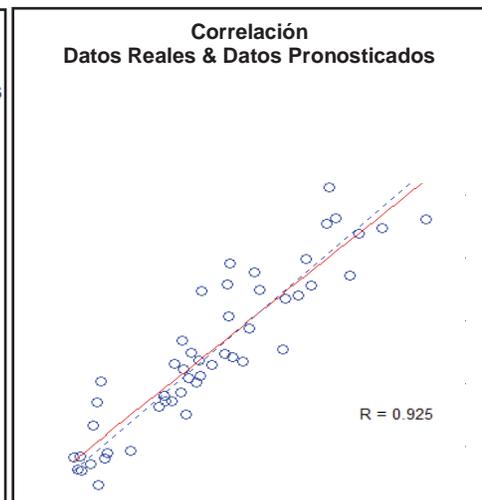
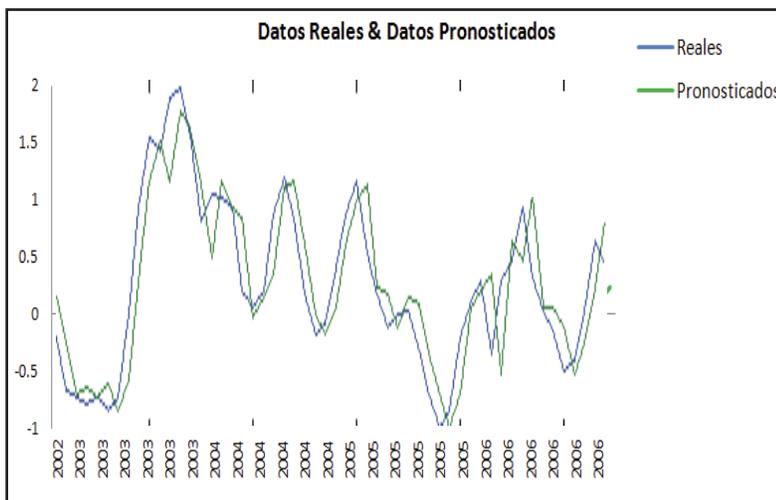


Figura 7.4 Gráfica comparativa mejor resultado general (Munivariado).

Figura 7.5 Correlación del mejor resultado.

8 CONCLUSIONES

De las tareas planteadas para el cumplimiento de los objetivos del trabajo, aparece concluida la elaboración del Estado del Arte, en el que se realiza la recopilación bibliográfica asociada al modelo de Redes Neuronales Recurrentes y de su algoritmo de aprendizaje basado en Poblaciones de Enjambre de Partículas (PSO), junto con el estudio de técnicas para seleccionar variables cuando el modelo es multivariado. El marco teórico profundizó el estudio de las variantes de PSO a experimentar, así como las métricas destinadas a evaluar la bondad del modelo y el proceso algorítmico que filtra aquellas variables más relevantes al pronóstico para mejorar su calidad.

El proceso de construcción de las soluciones respetó una metodología que consideró comenzar con el modelo univariado y luego con el multivariado, lo anterior permite comprobar si el costo de incorporar variables externas al modelo benefician al resultado del pronóstico. La experiencia obtuvo un mejor resultado al incorporar variables externas con un R^2 **Multivariado** de **0.8566** (**topología neuronal: 9 Entradas, 7 Nodos Ocultos, 40 Partículas, 1000 Iteraciones**) respecto al R^2 **Univariado** de **0.8470** (**topología neuronal: 7 Entradas, 7 Nodos Ocultos, 40 Partículas, 600 Iteraciones**), esta diferencia menor indica que la variable interna, Captura, posee la información relevante del pronóstico. La topología del modelo neuronal de este resultado también está evidenciado con el cálculo de relevancias correlativas, donde **Captura** supera considerablemente esta medición, en sus primeros **6 desfases**, a las variables TSM - **desfase 7** y **Niño34 - desfase 24**, aún siendo éstas dos últimas las seleccionadas como relevantes y no redundantes en el mejor resultado del modelo multivariado (CA_{t-1} a $t-6$, TSM_{t-7} , $N34_{t-24}$).

Con respecto al proceso de selección de variables, se comprobó que la inclusión de un proceso previo de **filtro** da como resultado una disminución drástica de la dimensionalidad de los datos, asegurando buenos resultados y minimizando el desgaste de la experimentación. Se analizó la relación lineal entre variable interna y externas en base a dependencia **correlativa** y el comportamiento no lineal estimando la **información mutua** entre ellas, resultando la relación lineal la predominante en el mejor pronóstico. Por lo anterior, si bien el mejor resultado en multivariado superó el univariado, se destaca que aplicar la metodología teórica de selección de variables deja un valor agregado al conocimiento práctico de soluciones predictivas con variables externas.

Similar conclusión arrojó la aplicación de variantes del aprendizaje PSO. El mejor resultado correspondió a PSO con $R^2 = 0.8566$, pero con un margen de diferencia no significativa a su variante DAPSO con $R^2 = 0.8533$. El algoritmo IPSO también se ubicó en los mejores resultados con $R^2 = 0.8490$,

lo cual determina que para este set de datos, la optimización por enjambre puede aplicarse en sus tres modos logrando buen pronóstico.

En relación a la evaluación final del mejor resultado, se logró un modelo neuronal que puede pronosticar satisfactoriamente la variabilidad del comportamiento del sistema en estudio; esto da cumplimiento a la necesidad planteada inicialmente y que corresponde a entregar un elemento de gestión que permite diseñar mejores políticas y determinar cuotas de pesca de Anchovetas, optimizando las estrategias del desarrollo de la industria pesquera en el norte de nuestro país.

Finalmente, se puede concluir que esta experiencia de desarrollo cumple sus expectativas al obtener como producto un modelo neuronal de pronóstico de captura de una especie pelágica situada entre las de mayor industrialización nacional, además de un plan metodológico definido y operativo con funciones automatizadas para pronóstico de solución multivariada aplicable a cualquier ítem de datos de interés.

9 BIBLIOGRAFÍA

1. **Aliaga R., Bernardo, Gómez U., Daniel y Neira A., Sergio.** *Análisis bioeconómico de la pesquería de sardina (*Sardinops sagax*) y anchoveta (*Engraulis ringens*) de la zona norte de Chile.* Valparaíso : s.n., 2001. 0717-7178.
2. **Espíndola, Fernando, Yáñez, Eleuterio y Böhm, Gabriela.** *Estandarización de las Tasas de Captura de Recursos Pelágicos de la Zona Norte de Chile.* 2005.
3. **Engelbrecht, Andries P.** *Computational Intelligence An Introduction.* Second Edition. Pretoria : Wiley, 2007.
4. **Llano, Luis, y otros.** *Comparison of the Activation Functions Performance in Neural Networks Feedforward to approximate Data Functions with and without Noise.* s.l. : Escuela de Ingeniería de Sistemas, Universidad Nacional de Colombia Sede Medellín, 2007.
5. **Haykin, Simon.** *Neural Networks: a Comprehensive Fundation.* Ontario : Pearson Pentice Hall, 2005. 9788178083001.
6. **Bertona, Luis Federico.** *Entrenamiento de Redes Neuronales basado en Algoritmos Evolutivos.* Buenos Aires : Universidad de Buenos Aires, 2005.
7. **Van Der Stockt, Stefan.** *A Generic Neural Network Framework Using Design Patterns.* Computer Intelligence Research Group. Pretoria : University of Pretoria, 2007.
8. **Palacios González, Federico.** *Redes Neuronales Artificiales y Métodos de Estimación de Funciones.* Universidad Politécnica de Valencia. Oviedo : Anales de Economía Aplicada. XIV Reunión ASEPELT, 2000.
9. **Quezada Román, Pablo.** *Redes RRBFF con Aprendizaje Híbrido para la Predicción de Captura de Anchovetas.* s.l. : Pontificia Universidad Católica de Valparaíso, 2009.
10. **Qian, Minglun.** *Neural network learning for time-series predictions using constrained formulations.* Illinois : University of Illinois at Urbana-Champaign, 2005.

11. **Matich, Damián Jorge.** *Redes Neuronales: Conceptos Básicos y Aplicaciones.* Rosario : Universidad Tecnológica Nacional, 2001.
12. **García Nieto, José Manuel.** *Algoritmos Basados en Enjambre de Partículas Para la Resolución de Problemas Complejos.* Málaga : Universidad de Málaga, 2006.
13. **Pérez López, Jesús Ramón.** *Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo.* Santander : Universidad de Cantabria, 2005.
14. **Cervantes Rovira, Alejandro.** *Clasificación mediante Enjambre de Prototipos.* Leganés : Universidad Carlos III de Madrid, 2009.
15. **Paggi, Horacio.** *Predicción de Series Temporales utilizando Redes Neuronales.* Montevideo : Facultad de Ingeniería, Universidad de la República, 2003.
16. **Verrastro, Claudio, Alberino, Sergio y Folino, Pablo.** *Control PID con esquema adaptivo de filtrado de ruido.* Buenos Aires : Grupo de Inteligencia Artificial UTN-FRBA, 2005.
17. **Bin Jiao, Zhigang Lian, Xingsheng Gu.** *A dynamic inertia weight particle swarm optimization algorithm.* Shanghai : Research Institute of Automation, East China University of Science and Technology, 2006.
18. **Xueming Yang, Jinsha Yuan, Jiangye Yuan, Huina Mao.** *A modified particle swarm optimizer with dynamic adaptation.* s.l. : Department of Electronic and Communication Engineering, North China Electric Power University, 2007.
19. **Hanke, John E.** *Pronósticos en los Negocios, 8va Edición.* México : Pearson Educación, 2006.
20. **SERNAPESCA,** *Anuarios Estadísticos de Pesca. Servicio Nacional de Pesca, Chile.*
21. **Thangaraj, Radha, y otros.** *Hybrid Evolutionary Algorithm for Solving Global Optimization Problems.* France : s.n.
22. **Rodríguez, Nivaldo, Gutiérrez-Estrada, Juan Carlos y Silva, Claudio.** *Monthly catch forecasting of anchovy *Engraulis ringens* in the north area of Chile: Non-linear univariate approach.* s.l. : ScienceDirect, 2007.

23. **Webster, Allen L.** *Estadística Aplicada a los Negocios y la Economía, Tercera Edición.* s.l. : McGraw-Hill, 2000.
24. **Reig, Lucas Pujol.** *Predicción de Caudales en tiempo real en grandes cuencas utilizando Redes*
25. **Rodríguez, Nibaldo, Gutiérrez-Estrada, Juan, Yáñez, Eleuterio, Plaza, Francisco, Barbieri, M.A., Pulido-Calvo, Inmaculada, Bórquez, Cinthya.** *Anchovy and Sardine abundance forecast off northern Chile: A Multivariate ecosystemic neural network approach, Progress in Oceanography, www.elsevier.com, 2010*
26. **Cesar Augusto Cardona, Juan David Velásquez.** *Selección de Características Relevantes usando Información Mutua:* Dyna, Universidad Nacional de Colombia, 2006.
27. **Ismael Lapiedra Michel.** *Desarrollo de Algoritmos de selección de variables para clasificadores neuronales:* Universitat Rovira I Virgili, 2004
28. **Andreia Dionísio, Rui Menezes, Diana A. Mendes.** *Mutual Information: a dependence measure for nonlinear time series:* University of Évora Portugal, 2004
29. **José Luis Torrecilla Noguerales.** *Análisis de Datos Funcionales, Clasificación y Selección de Variables:* Universidad Autónoma de Madrid, 2010
30. **Irene Rodríguez Luján.** *Selección de variables mediante programación cuadrática:* Universidad Autónoma de Madrid, 2009
31. **Andreia Dionisioa, Rui Menezesb, Diana A. Mendesb.** *Mutual information: a measure of dependency for nonlinear time series, Physica A 344 (2004) 326–329, www.elsevier.com, 2004*
32. **Farshid Keynia.** *Engineering Applications of Artificial Intelligence, journal homepage: www.elsevier.com/locate/engappai, www.elsevier.com, 2012*