

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

**PERCEPTRON WAVELET RECURSIVO Y
ALGORITMOS GENÉTICOS
PARA LA PREDICCIÓN DE CAPTURA DE LA
ANCHOVETA**

RODRIGO DAVID BARAHONA LLORE

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

JUNIO, 2014

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

**PERCEPTRON WAVELET RECURSIVO Y
ALGORITMOS GENÉTICOS
PARA LA PREDICCIÓN DE CAPTURA DE LA
ANCHOVETA**

RODRIGO DAVID BARAHONA LLORE

PROFESOR GUÍA: **NIBALDO RODRIGUEZ**

CARRERA: INGENIERÍA CIVIL EN INFOMÁTICA

JUNIO, 2014

A mi Familia.

Contenidos

Contenidos	1
Glosario de términos	3
Lista de figuras	4
Lista de tablas	5
1. Introducción	7
1.1. Introducción.....	7
1.2. Objetivos.....	10
1.2.1. Objetivos Generales	10
1.2.2. Objetivos Específicos	10
1.3. Organización del texto	10
2. Redes Neuronales	13
2.1. Introducción.....	13
2.2. Redes Neuronales biológicas	14
2.3. Redes Neuronales artificiales	15
2.4. Características de las RNA.....	16
2.4.1. Aprendizaje Adaptativo	16
2.4.2. Auto-organización.	16
2.4.3. Tolerancia a Fallos	17
2.4.4. Operación en tiempo real	17
2.5. Topologías de redes neuronales	17
2.5.1. Redes de Propagación hacia Adelante (Feed-Forward).....	17
2.5.2. Redes de Retro-Propagación (Feed-Back).....	20
2.5.3. Funciones de activación.....	22
2.6. Aprendizaje de una red neuronal	30
2.6.1. Aprendizaje supervisado	30
2.6.2. Aprendizaje no supervisado.....	32
3. Algoritmos Genéticos	33
3.1. Introducción.....	33
3.2. Reseña Histórica	34
3.3. Operadores de un Algoritmo Genético	34
3.3.1. Cromosomas	35

3.3.2.	Presión selectiva del medio	35
3.3.3.	Cruce de Cromosomas	35
3.3.4.	Mutación de cromosomas	35
3.4.	Algoritmo Genético Simple.....	36
3.5.	Codificación del cromosoma.....	37
3.5.1.	Codificación Binaria	38
3.5.2.	Codificación real y alfabética	39
3.6.	Evaluación y Selección.....	39
3.7.	Aplicaciones de Algoritmos Genéticos	41
4.	Predictor Neuro-genetico	43
4.1.	Algoritmos genéticos y pesos de la red	43
4.2.	Determinando la Topología de la red	44
4.3.	Modelo de funcionamiento de la Red	45
4.4.	Desempeño de una RNA	46
4.5.	Elección de padres	47
4.6.	Generar nueva población	47
5.	Discusión de Resultados.....	49
5.1.	Introducción.....	49
5.2.	Procedimiento.....	50
5.3.	Análisis de convergencia y configuración.....	50
5.3.1.	Graficas Obtenidas Modelos Elman y Jordan	52
5.3.2.	Modelo de Jordan	58
5.3.3.	Modelo de Elman	59
5.4.	Comparativa de Modelos Predictivos	61
5.4.1.	Tiempo de Ejecución y Cantidad de Parámetros.....	62
5.4.2.	Coefficiente de Determinación R^2	62
5.4.3.	Interpretación del resultado	62
5.4.4.	Ganancias de los modelos propuestos.....	63
6.	Conclusión.....	64
7.	Referencias	66

Glosario de términos

RNA	:	Red Neuronal Artificial
RBF	:	Red De Base Radial
PM	:	Perceptron Multicapa
TW	:	Transformada Wavelet
WG	:	Wavelet Gaussiana
CWT	:	Transformada Continua Wavelet
DTW	:	Transformada Discreta Wavelet
TF	:	Transformad De Fourier
ART	:	Redes De Resonancia Adaptativa
AG	:	Algoritmo Genético
ADN	:	Ácido Desoxirribonucleico
CF	:	Crowding Factor
UPS	:	Unidad práctica de salinidad
RNW	:	Red Neuro-Wavelet
RMSE	:	Error cuadrático medio
R ²	:	Coefficiente de determinación
UPS	:	Unidades Prácticas de Salinidad
MT	:	Millones de Toneladas

Lista de figuras

Figura 1-1 : Gráfico desembarco de anchovetas.	8
Figura 2-1 : Neurona Biológica	14
Figura 2-2 : Modelo de neurona Artificial	16
Figura 2-3 : Red perceptron multicapa	18
Figura 2-4 : Estructura de una Red RBF.....	19
Figura 2-5 : Red neuronal de Jordan.....	20
Figura 2-6 : Red neuronal de Elman.....	21
Figura 2-7 : Función de Activación Sigmoidal.....	22
Figura 2-8 : Wavelet Haar.....	24
Figura 2-9 : Wavelet Mexican Hat.....	25
Figura 2-10 : Wavelet Daubechies de segundo orden	25
Figura 2-11 : Wavelet Morlet	26
Figura 2-12 : Wavelet Gaussiana	27
Figura 2-13 : Aprendizaje Neuronal Supervisado	31
Figura 2-14 : Aprendizaje Neuronal No Supervisado	32
Figura 3-1 : Operador de Cruce	38
Figura 4-1 : Cromosoma de Contexto	44
Figura 4-2 : Cromosoma de Pesos	44
Figura 4-3 : Cromosoma Final.....	44
Figura 4-4 : Diagrama De Bloques RNA.....	45
Figura 4-5 : Diagrama De Bloques Rna Con Filtro De Suavizado	46
Figura 5-1 : Grafico De Convergencia Modelo De Jordan	51
Figura 5-2 : Grafico De Convergencia Modelo De Elman	51
Figura 5-3 : Grafico De Convergencia Entre Modelos Elman Y Jordan	52
Figura 5-4 : Grafico De Convergencia Modelo De Jordan 100p-200g.....	52
Figura 5-5 : Grafico De Dispersión Modelo De Jordan 100p-200g	53
Figura 5-6 : Grafico De Convergencia Modelo De Jordan 200p-200g.....	53
Figura 5-7 : Grafico De Dispersión Modelo De Jordan 200p-200g	53
Figura 5-8 : Grafico De Convergencia Modelo De Jordan 300p-200g.....	54
Figura 5-9 : Grafico De Dispersión Modelo De Jordan 300p-200g	54
Figura 5-10 : Grafico De Convergencia Modelo De Jordan 400p-200g.....	54
Figura 5-11 : Grafico De Dispersión Modelo De Jordan 400p-200g.....	55
Figura 5-12 : Grafico De Convergencia Modelo De Elman 100p-200g.....	55
Figura 5-13 : Grafico De Dispersión Modelo De Elman 100p-200g	55
Figura 5-14 : Grafico De Convergencia Modelo De Elman 200p-200g.....	56
Figura 5-15 : Grafico De Dispersión Modelo De Elman 200p-200g	56
Figura 5-16 : Grafico De Convergencia Modelo De Elman 300p-200g.....	56
Figura 5-17 : Grafico De Dispersión Modelo De Elman 300p-200g	57
Figura 5-18 : Grafico De Convergencia Modelo De Elman 400p-200g.....	57
Figura 5-19 : Grafico De Dispersión Modelo De Elman 400p-200g	57

Lista de tablas

Tabla 4-1 : Codificación directa de la topología de una red neuronal	44
Tabla 5-1: Resultados Obtenidos Configuración 8in-10hid Para Jordan.....	58
Tabla 5-2 : Resultados Obtenidos Configuración 8in-11hid Para Jordan	58
Tabla 5-3 : Resultados obtenidos configuración 8in-12HID para Jordan	59
Tabla 5-4 : Resultados obtenidos configuración 8in-13HID para Jordan	59
Tabla 5-5 : Promedio De Las Pruebas Realizadas Al Modelo Jordan 200p-200g.....	59
Tabla 5-6 : Resultados Obtenidos Configuración 8IN-10HID para Elman	60
Tabla 5-7 : Resultados Obtenidos Configuración 8iN-11HID para Elman	60
Tabla 5-8 : Resultados Obtenidos Configuración 8in-12hid Para Elman.....	60
Tabla 5-9 : Resultados Obtenidos Configuración 8in-13hid Para Elman.....	61
Tabla 5-10 : Promedio De Las Pruebas Realizadas Al Modelo Elman 200p-200g.....	61
Tabla 5-11 – Comparativa De Modelos Propuestos	61
Tabla 5-12 – Comparativa De Ganancia de Modelos Propuestos	63

Resumen

En América del sur las especies pelágicas que cuentan con un mayor nivel de extracción son la anchoveta y la sardina además de ser las más numerosas. La alta producción de fitoplancton dada en la zona sur de Perú y norte de Chile, específicamente la primera y la segunda región, favorece a que el zooplancton, alimento de la anchoveta sea abundante, siendo esta zona por sus condiciones, especial para el crecimiento de esta especie. El porcentaje de extracción de especies pelágicas a 1996 en Chile, era aproximadamente el 20 % del total de las capturas mundiales, es por esto, que desde 1964 se han realizado diversos estudios, buscando desarrollar un modelo predictivo a usarse para la estimación de captura mensual de la anchoveta, recurso que será estudiado en este documento. En la actualidad, recientes investigaciones han demostrados que modelos pertenecientes al campo de la inteligencia artificial, muestran una mayor precisión y potencia que los modelos tradicionales para determinar el volumen mensual de extracción del recurso. Estos modelos también conocidos como heurísticos, encontramos los algoritmos genéticos y las redes neuronales, los cuales han sido utilizados en este proyecto de título para predecir el volumen de capturas mensuales de anchoveta en la zona norte de Chile. El mejor modelo de predicción encontrado está formado con 8 nodos input, 12 nodos ocultos y un nodo de salida. Este modelo neuronal obtuvo una varianza explicada de 91,45% con un MAPE de 20,15%.

Abstract

The most important pelagic species in South America's by their abundance are anchovy and sardine. The given high phytoplankton production in southern Peru and northern Chile, specifically the first and the second region, favors the zooplankton, anchovy food is plentiful, with this area for its special conditions for the growth of this species. The percentage extraction of pelagic and 1996 in Chile was approximately 20% of total world catch, for these reason since 1964 there have been several studies, seeking to develop a predictive model to be used for estimating monthly catch anchovy. Recent researches has shown that models belonging to the field of artificial intelligence, show a higher accuracy than traditional models power to determine the volume of extraction. Within these models also known as heuristics, are genetic algorithms and neural networks to be used in determining forecasting models of these species. So with a mixture of both managed to get two models to help us in the next month's estimate extraction anchovies, these models are evaluated and compared with the actual data is to see how close they are to reality. Once certain models and coefficients which indicate how accurate the model proceeds to buy these with other proposed models. A real approximation to the desired solution can be obtained using more than one model and thus by averaging errors can determine more accurate are the steps to extract this pelagic resource. These models also known as heuristics, we found genetic algorithms and neural networks, which have been used in this project to predict the volume of monthly anchovy catches in northern Chile. The best prediction model found, is formed with eight input nodes, 12 hidden nodes and an output node. This neural model, obtained an explained variance of 91.45%, with a MAPE of 20.15%.

1. Introducción

1.1. Introducción

La anchoveta es una especie pelágica, de talla pequeña, que puede alcanzar hasta los 20 cm de longitud total, vive en aguas moderadamente frías, con rangos que oscilan entre 16° y 23°C en verano y de 14° a 18°C en invierno; La salinidad de esta agua puede variar entre 34.5 y 35.1 UPS. La anchoveta tiene hábitos altamente gregarios formando enormes y extensos cardúmenes que en periodos de alta disponibilidad, facilita que sus capturas sean de gran magnitud.

En América del sur la extracción de especies pelágicas es de un volumen considerable, es por esto que se busca medir y racionar la extracción de estas, de manera de no afectar el ecosistema propio de la zona norte del país. Debido a su abundancia, las especies con mayor nivel de extracción son la anchoveta y la sardina. Dentro de los factores que afectan al ecosistema marino de estas especies, se encuentran las corrientes de aguas cálidas presentes en la superficie marina, esto contribuye a que sardinias y anchovetas profundicen con un cierto límite, (*a mayor profundidad, mayor es la concentración salina y menor la concentración de zooplancton*), a zonas en donde su extracción con red de cerco es dificultosa, y /o a zonas como el sur de Perú, y la zona norte de Chile, en donde el efecto de estas corrientes mengua. La alta producción de fitoplancton dada en la zona sur de Perú y norte de Chile, específicamente la primera y la segunda región, favorece a que el zooplancton sea abundante, ninguna otra zona es tan fértil como esta para el crecimiento de la anchoveta.

El porcentaje de extracción de especies pelágicas a 1996 en Chile era aproximadamente el 20 % del total de las capturas mundiales, es por esto que desde 1964 se han realizado diversos estudios y cruceros bio-oceanográficos estacionales, para determinar el comportamiento de este recurso, logrando identificar ciertos índices, como es el de densidad de los cardúmenes y cobertura geográfica.

Estos índices sumados además a un comportamiento dado por ciertas corrientes marinas, han determinado que en el talud continental de Chile existe una compleja estructura de variabilidad [1], que esta forzada desde el Pacífico Ecuatorial, por las oscilaciones de Madden-Julian en la banda intra-estacional [2], por vientos ecuatoriales [3], El Niño [4], y la oscilación del sur, que en conjunto abren un potencial de predictibilidad de las condiciones oceanográficas con algunos meses de anticipación.

La información disponible sobre variables, corrientes, fluctuaciones y cambios estacionales, describen principalmente algunas relaciones funcionales entre el ambiente y el recurso pelágico, pero no se puede determinar con certeza la dinámica de las poblaciones debido ya que aún se desconocen todos los procesos y mecanismos del ambiente marino. Se busca por ende un modelo que integre tanto variables oceanográficas, biológicas, y pesqueras, orientado principalmente al apoyo en la toma de decisiones en la administración pesquera sobre el uso de sus recursos.

En la economía Chilena al 2008, las exportaciones son lideradas por el sector minero con un 68,1%, siguiendo con las exportaciones industriales con un 20,2%, en tercer lugar con un 5,8 % se encuentra la pesca, dentro de este último se tiene que la extracción de anchoveta ocupa gran parte de este porcentaje (cerca del 34%), es por esto que fluctuaciones en la extracción de este recurso se ven traducidas inmediatamente en un cambio en el índice de actividad económica en las primeras regiones, reduciendo ingresos a la región en materia de impuestos y aumentando el índice de desempleo, el restante de las exportaciones industriales pesqueras se encuentran otras especies pelágicas tales como la caballa, el jurel y la sardina [5].

Estos cambios en la economía deben ser previstos, de manera de que las autoridades chilenas sepan enfrentar la situación dada y estén preparadas para los distintos escenarios posibles, tales como la disminución del recurso pelágico, es por esto que el objetivo principal de la predicción es ayudar en este punto ya que brinda indicadores que se pueden utilizar en la toma de decisiones tales como la concentración de recursos en un lugar y en un determinado momento, por un determinado de tiempo, para lograr una mejor distribución de los recursos. Para que la predicción sea válida es necesario contar con datos históricos, información cualitativa y cuantitativa de la pesquería chilena, asumiendo que ciertos comportamientos se repetirán en el futuro.

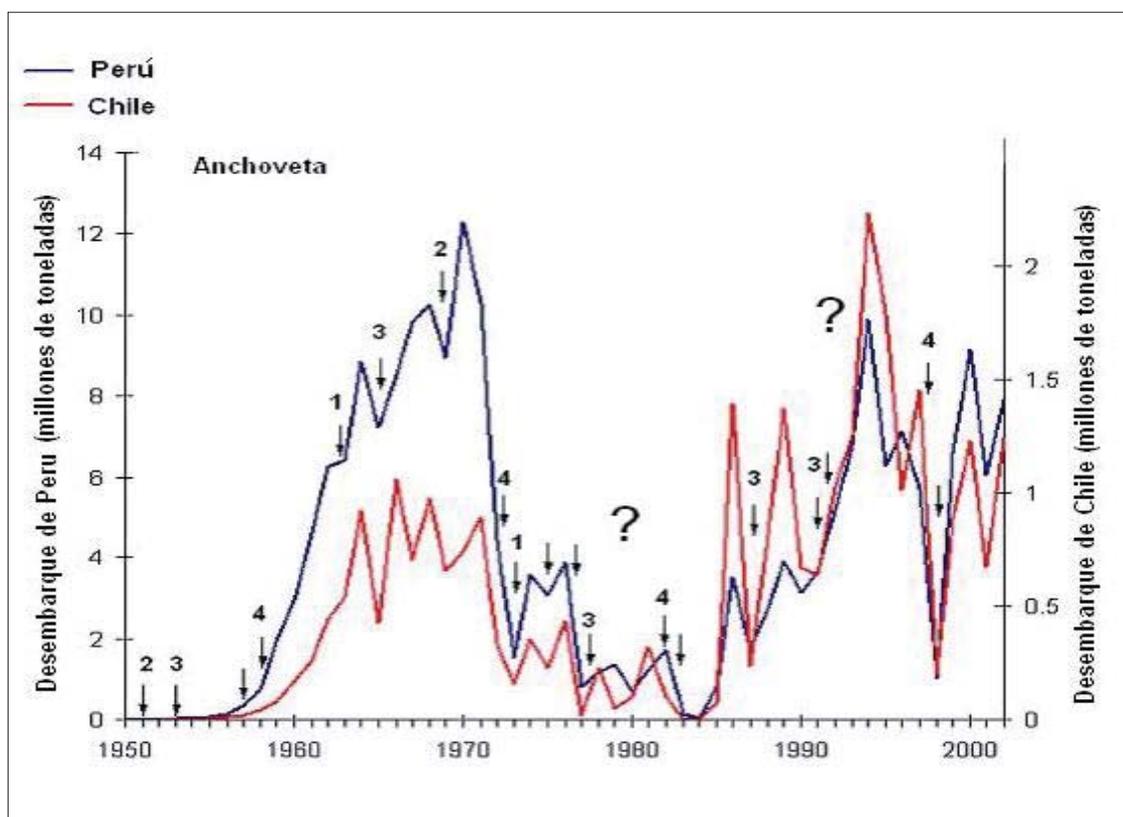


FIGURA 1-1 : GRÁFICO DESEMBARCO DE ANCHOVETAS.

La figura 1-1, muestra el volumen de extracción de anchovetas desde los años 1950 hasta comienzos del año 2000, estos datos han sido recolectados por SernaPesca, entidad gubernamental chilena que corresponde al “*Servicio Nacional De Pesca Y Agricultura*”, de

esta se infiere que los datos recolectados son dispersos y no están definidos por una función conocida (curva conocida), teniendo muchos picos, lo que traduce cambios drásticos en los niveles de extracción debido a la explotación indiscriminada de este recurso en años anteriores. En base a estos datos se diseñan políticas, estrategias, medidas de administración y de contención de forma de evitar que cambios bruscos produzcan un efecto negativo en la economía del país, así no solo se tiene en cuenta aspectos financieros, sino que factores como biológicos, técnicos, sociales, legales, que colaboran a la determinación del correcto funcionamiento de la entidad pesquera. Esta última es la encargada de generar la información histórica que se utilizara para el diseño e implementación de cambios en los planes de manejo de las pesquerías chilenas. La información recopilada en estos años ha servido para que además de los desafíos ya establecidos se impongan nuevos desafíos, se busquen nuevos modelos predictivos, más certeros, favoreciendo la economía del país además de facilitar la toma de decisiones de la administración pesquera.

Para poder desarrollar algunos de estos modelos predictivos se utilizan técnicas analíticas, tales como aproximaciones ecológicas, análisis de cohortes, además de técnicas en base determinista como regresiones lineales y no lineales, técnicas uní-variadas y multi-variadas utilizando gran variedad de técnicas estadísticas para la predicción en pesquerías. En la actualidad, recientes investigaciones han demostrado que modelos pertenecientes al campo de la inteligencia artificial, muestran una mayor precisión sobre los modelos de predicción tradicionales. Dentro de estos modelos también conocidos como heurísticos, se encuentran los modelos de lógica difusa, algoritmos genéticos, y redes neuronales, que trataremos en este documento.

Las ventajas de estos modelos frente a modelos convencionales están en la capacidad de poder modelar sistemas no lineales y no equilibrados, para poder hacer esto se incurre en una transformación no lineal de los datos de entrada y salida, supone esto ventajas frente a metodologías convencionales en donde la relación de los datos siempre ha de ser lineal. En adelante se utilizara un algoritmo de red neuronal para pronosticar la extracción de anchovetas en la zona norte de Chile denominado Wavelet.

Al ser una red neuronal artificial posee la capacidad de “*aprender de la experiencia*”, así mediante la RNA Wavelet (*familia de funciones*), se puede combinar variables como tiempo, frecuencia, permitiendo llegar a una solución más precisa, de acuerdo a los datos de entrada y de salida.

1.2. Objetivos

1.2.1. Objetivos Generales

Desarrollar un modelo predictivo para determinar el volumen de captura mensual de Anchovetas en la zona norte de Chile utilizando redes neuronales recurrentes Wavelet y Algoritmos genéticos.

1.2.2. Objetivos Específicos

- Explicar las estructuras y funcionamiento de las redes neuronales recurrentes evolutivas.
- Diseñar y estimar los parámetros del predictor Neuro-evolutivo para los modelos propuestos de Jordan y Elman.
- Evaluar el rendimiento de los predictores neuronales de Elman y Jordan utilizando métricas residuales.

1.3. Organización del texto

El Capítulo 2, hace referencia a las Redes Neuronales, esta se compone de 3 secciones, en la primera de esta sección 2.1 Introducción, se hace una referencia histórica de las redes neuronales, pasando por la importancia que estos han tenido a través del tiempo hasta las características que los conforman. En la sección 2.2, redes neuronales biológicas, nos dan las características que son, la forma en que trabajan, su capacidad de aprendizaje, su topología, viéndose esto reflejado en la sección 2.3 en donde se habla de las RNA artificiales, en donde se emula una RNA biológica, adaptándola a un problema específico, simulando las mismas características de una RNA biológica, dotando a esta de una topología específica, de funciones de activación acotadas, de forma de lograr que esta última tenga además la capacidad de aprender características descritas en la sección 2.4. Es así como en la sección 2.5, se pueden describir distintas topologías asociadas a la RNA. Por topología se entiende la configuración de la red, de sus interconexiones neuronales y de la cantidad de capas presentes en la red. El hablar de las características, de la topología y de su parecido a la RNR biológica, nos lleva a hablar también de la característica más importante de una red neuronal, la capacidad de aprendizaje, de esto se hace referencia en la sección 2.6, en donde se explica la diferencia entre aprendizaje supervisado y el no supervisado, dependiendo de cómo se quieran medir los resultados obtenidos.

En el tercer capítulo de este apartado, se hace referencia a los algoritmos genéticos, comenzando en la sección 3.1 con una introducción al tema. En la sección 3.2, se da una reseña histórica y de su utilización para resolver problemas de la cotidianeidad. En la sección 3.3 se habla sobre los operadores de un AG de forma de describir sus componentes, manejo de estos y como se desarrollan nuevas generaciones de individuos el cruce entre estos y la variación que tienen creando nuevas poblaciones, describiendo el funcionamiento de estos en la sección 3.4 en su expresión más simple. Es así como en la sección 3.5 se describe la forma en que el AG es codificado, de forma de poder modelarlo. En la sección 3.6, se hace referencia a la forma en que se eligen los individuos para la formación de una nueva población, los métodos de selección y por último se hace una segunda referencia histórica dando ejemplos de

la aplicación de los AG en la sección 3.7, dejando en claro que estos son aplicables casi a cualquier problema pensado por el hombre.

Una vez que se tiene clara la forma en que trabajan los AG y las RNA, en el capítulo 4 se propone un modelo de predicción de respuesta, basado en ambos para poder converger de mejor forma a una solución determinada es para esto que en la sección 4.1 se habla sobre los AG y determinar los pesos de la red, así también se determina en la sección 4.2, la topología de red que se utilizara, determinando en la sección 4.3 el modelo en que funcionara este algoritmo Neuro-genético propuesto. Una vez que se tiene definido el modelo y la configuración que este tendrá se debe determinar cuál es la efectividad del modelo propuesto es por esto que en la sección 4.4 se revisan como se evaluara el desempeño del algoritmo con esto se puede tener en la sección 4.5 claridad de cómo se elegirá la sucesión o padres para con estos generar una nueva población descrito en la sección 4.6.

Esta evaluación de los resultados descrita en el capítulo anterior se hace en el capítulo 5, partiendo con una pequeña introducción en la sección 5.1, pasando al procedimiento que se utilizara, para la evaluación del modelo en la sección 5.2, En la sección 5.3 se analiza la convergencia que tuvo el algoritmo con las distintas configuraciones propuestas, de forma de poder determinar cuál de las configuraciones propuestas es la mejor, se debe hacer una comparativa de los modelos en cada una de sus configuraciones, esto es hecho en la sección 5.4.

Una vez obtenidos los resultados en el capítulo anterior en el capítulo 6 se rescataran las características de los algoritmos Neuro-recurrentes implementados llevándolas a la casuística propuesta de forma inicial, la determinación del volumen de captura mensual de anchovetas en la zona norte del país.

Este proyecto de título se basó en estudios realizados con anterioridad por distintos autores es así como en el capítulo 7 se hace referencia a los documentos utilizados a lo largo de la investigación.

1. Capítulo 2: Referido a las Redes Neuronales, en donde se hace una reseña histórica y explicando la importancia que estos tienen, detallando además los tipos de redes, sus características, las distintas topologías que existen, funciones de activación más comunes y como las neuronas tienen la capacidad de aprender.
2. Capítulo 3: En este se habla sobre los algoritmos genéticos y de su utilidad, de cómo este puede ayudar en la calibración de una red, la forma en que trabaja y cuáles son sus componentes.
3. Capítulo 4: En capítulos anteriores se describieron de redes neuronales y de algoritmos genéticos, en este capítulo se habla de cómo estos dos algoritmos se pueden fusionar y generar un nuevo algoritmo neuro-genético.
4. Capítulo 5: En este capítulo se describe los resultados de la aplicación del algoritmo neuro-genético generado en el capítulo 4. Los resultados son presentados de manera gráfica y en tablas de resultados, analizando desempeño, convergencia, de forma de analizar qué tan alejado de la realidad resulto el modelo propuesto realizando una comparativa con otros modelos propuestos.

5. Capítulo 6: Se hace referencia al resultado obtenido en el capítulo anterior sobre los coeficientes de correlación y de determinación, de forma de ver que tan preciso fue el modelo y cuanto nos ayudó en la predicción de la captura de anchoveta.
6. Capítulo 7: En este, se incluyen las referencias utilizadas y que hicieron posible este documento.

2. Redes Neuronales

2.1. Introducción

Desde el inicio de los tiempos el hombre siempre ha buscado superar barreras, obstáculos, descubrir, estudiar, comprender e innovar con el fin de simplificar la vida cotidiana. Para desarrollar cada una de estas tareas es necesario tiempo y esfuerzo, es por esto que ante la necesidad de reducir estos, se busca la forma de automatizar ciertas tareas, para esto, se crearon las computadoras, maquinas capaces de realizar cálculos matemáticos complejos, en cortos periodos de tiempo, convirtiendo problemas considerados antes muy engorrosos, en una simple tarea. Existen además ciertas tareas que aunque simples para las personas, no obedecen a patrones o a un planteamiento algorítmico y que para un computador serian notablemente difíciles de realizar, como en algún momento lo fue el reconocimiento facial.

Se tiene así un nuevo desafio, hacer que a través de computadoras se emule una red neuronal biológica, para así poder resolver este nuevo tipo de problemas. Para esto las maquinas debían imitar algunas características importantes que posee esta red como son:

1. **Aprendizaje inductivo:** No se le indican las reglas a la red para dar una solución, sino que extrae sus propias reglas a partir de los ejemplos de aprendizaje (entrenamiento), modifica su comportamiento en función de la experiencia. Esas reglas quedan almacenadas en las conexiones y no representadas explícitamente como en los sistemas basados en conocimiento (simbólico-deductivos).
2. **Generalización:** Una vez se tiene la red entrenada, se le pueden utilizar distintos datos a los del aprendizaje, la respuesta obtenida dependerá del parecido de los datos con los ejemplos de entrenamiento.
3. **Abstracción o tolerancia al ruido:** Las redes neuronales artificiales son capaces de extraer o abstraer las características esenciales de las entradas aprendidas, de esta manera pueden procesar correctamente datos incompletos o distorsionados.
4. **Procesamiento paralelo:** Las neuronas reales trabajan en paralelo; En el caso de las redes artificiales es obvio que si usamos un solo procesador no podrá haber proceso paralelo real, sin embargo hay un paralelismo inherente, lo esencial es que la estructura y modo de operación de las redes neuronales las hace especialmente adecuadas para el procesamiento paralelo real mediante multiprocesadores (se están desarrollando máquinas específicas para la computación neuronal).
5. **Memoria distribuida:** El conocimiento acumulado por la red se halla distribuido en numerosas conexiones, esto tiene como consecuencia la tolerancia a fallos: una red neuronal es capaz de seguir funcionando adecuadamente a pesar de sufrir lesiones con destrucción de neuronas o sus conexiones, ya que la información se

halla distribuida por toda la red, sin embargo en un programa tradicional un pequeño fallo en cualquier punto puede invalidarlo todo y dar un resultado absurdo o no dar ningún resultado.

Esta emulación es más conocida como redes neuronales artificiales (RNA), y serán explicadas con mayor profundidad en el Capítulo 2.3.

Existen diferentes tipos de redes neuronales artificiales diferenciándose entre sí por la forma en que están estructuradas y del modo en que realizan el aprendizaje, así dependiendo del problema que se pretende resolver, se puede elegir desarrollar la solución con una u otra red siendo algunas RNA más adecuadas que otras, en la resolución de un determinado problema.

2.2. Redes Neuronales biológicas

El cerebro consta de un gran número de elementos que se encuentran altamente interconectados, llamados neuronas. Estas interconexiones se dan a través de señales químico-eléctricas, la *sinapsis* es el nombre usado para notar una comunicación entre neuronas, debido a la cantidad de neuronas presentes en el cerebro humano, aproximadamente 10^{11} , la cantidad de sinapsis por elemento están alrededor de 10^4 conexiones por elemento, lo que hace que sea considerada como una red de alta complejidad, y gran capacidad de procesamiento.

La neurona es una célula compuesta de *dendritas* que son las encargadas de captar las señales químico-eléctricas emitidas desde la neurona anterior y traspasarla al *soma*, allí es donde se encuentra el *núcleo* de la neurona. El componente de la neurona que se encarga de transmitir esta señal hacia el resto de la célula, es el *axón*, filamento alargado, que termina en ramificaciones llamado *terminación del axón* y es el que se conecta con las dendritas de la neurona siguiente. El axón se encuentra cubierto de *mielina*, sustancia que rodea la mayor parte de las fibras nerviosas y que permite una gran velocidad de transmisión de los impulsos nerviosos, así la neurona puede recibir señales, impulsos eléctricos procedentes de otras neuronas gracias a las dendritas luego esta emite impulsos de actividad eléctrica a lo largo del axón, que terminan finalmente en una sinapsis con otra neurona [6].

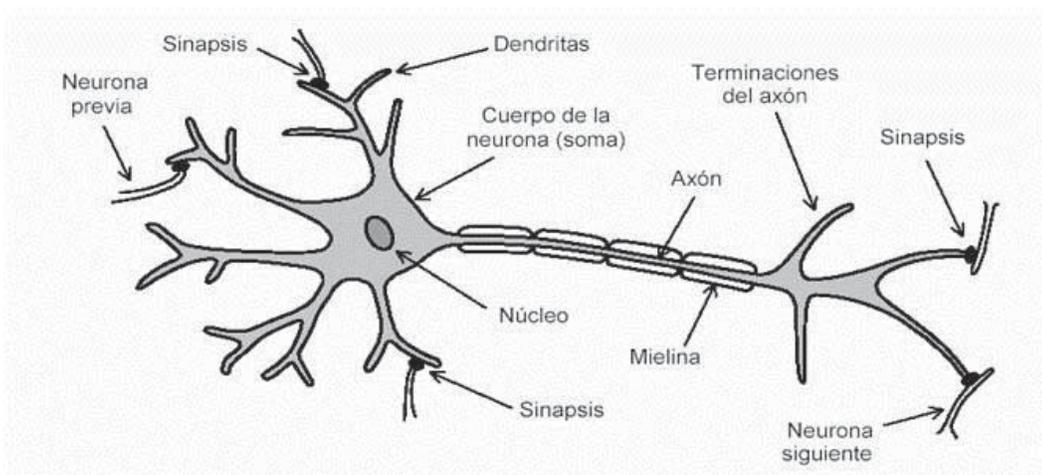


FIGURA 2-1 : NEURONA BIOLÓGICA

Algunas de las estructuras neuronales son determinadas en el nacimiento, otra parte es desarrollada a través del aprendizaje, proceso en que nuevas conexiones neuronales son realizadas y otras se pierden por completo. Las estructuras neuronales continúan cambiando durante toda la vida, estos cambios consisten en el refuerzo o debilitamiento de las uniones sinápticas; por ejemplo se cree que nuevas memorias son formadas por la modificación de esta intensidad entre sinapsis, así el proceso de recordar el rostro de un nuevo amigo, consiste en alterar varias sinapsis.

2.3. Redes Neuronales artificiales

Las RNA buscan imitar en pequeña escala el funcionamiento de las neuronas que conforman el cerebro humano. Warren McCulloch y Walter Pitts [7], dos investigadores dedicados al estudio de la neurofisiología propusieron un modelo matemático de neurona. En este modelo cada neurona estaba dotada de un conjunto de entradas y salidas, en donde las entradas tienen asociado un peso determinado, la función de activación de la neurona se calcula mediante la suma de los productos de cada entrada y la salida es una función de esta activación.

La principal clave de este modelo matemático de neurona se encuentra en los pesos de las diferentes entradas. Como se ha visto, las entradas son modificadas por el peso y las salidas son función de estas modificaciones, esto nos lleva a concluir que los pesos influyen de forma decisiva en la salida y por lo tanto pueden ser utilizados para controlar la salida que se desea, este proceso es llamado entrenamiento y consiste en aplicar entradas determinadas a la red y observar la salida que produce. Si la salida que produce no se adecua a la que se esperaba, se ajustan los pesos de cada neurona para interactivamente ir obteniendo las respuestas adecuadas del sistema, el cambio de pesos en las neuronas en la red es el aprendizaje de la neurona.

El modelo básico de la neurona artificial posee tres elementos [8]: Un grupo de *sinapsis* o *enlaces de interconexión*, esta caracteriza cada entrada a través de la multiplicación de la misma con un valor, el cual se denomina peso “ W ”, pudiendo adoptar valores positivos como negativos; un *estado interno* que se modifica a través de una combinación lineal con los parámetros de la señal de entrada, provenientes de la sinapsis de la neurona; y una *función de activación* encargada de limitar la amplitud de salida de la señal de la neurona, para así propagarlo al resto de la neurona. La amplitud de la señal de salida es normalizada en los intervalos $[0,1]$ y alternativamente $[-1,1]$ debido a que la función de activación que se utilizara corresponde a una del tipo sigmoidea.

El modelo de la figura 2-2 es la representación matemática de la neurona, aquí se puede apreciar la relación entre la sinapsis, el estado interno y función de activación, incluye una entrada externa que se denomina tendencia o *bias* y está denotada por b_k . Este parámetro adicional produce el efecto de añadir una cierta tendencia en los datos a la entrada de la función de activación y permite trasladar dicha función a la región de interés según sea la aplicación.

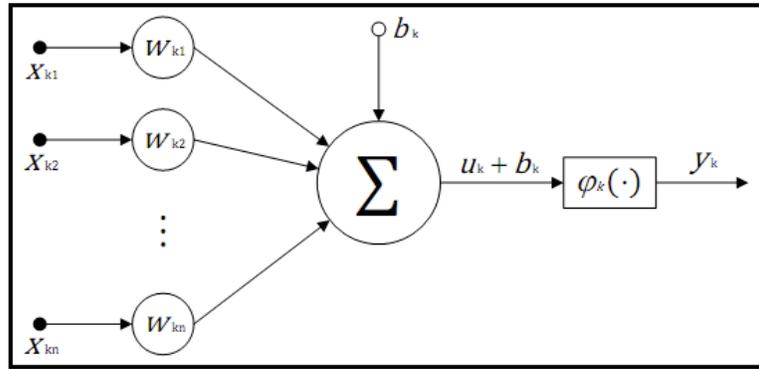


FIGURA 2-2 : MODELO DE NEURONA ARTIFICIAL

- La combinación lineal de las señales de entrada sopesadas por los respectivos pesos sinápticos, está dada por u_k , definida como:

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j \quad (2.1)$$

- El Potencial de activación de la neurona v_k y el bias b_k se define como:

$$v_k = u_k + b_k \quad (2.2)$$

- La señal de salida de la neurona está dada por y_k definida como $y_k = \varphi(v_k)$, donde $\varphi(\cdot)$ es la función de activación.
- Donde x_1, x_2, \dots, x_m son las señales de entrada, $w_{k1}, w_{k2}, \dots, w_{km}$ son los pesos sinápticos que unen la salida de la neurona k con las entradas de las neuronas $1, 2, \dots, m$.

2.4. Características de las RNA

Las RNA o redes neuronales artificiales, buscan imitar el comportamiento de las redes neuronales presentes en el cerebro humano, es por eso que poseen características comunes las cuales serán detalladas a continuación.

2.4.1. Aprendizaje Adaptativo

Las RNA, tienen la capacidad de aprender de los datos que se le presentan, puede establecer relaciones, por lo que no es necesario establecer un modelo a priori o especificar alguna función de probabilidad para el problema, esta capacidad está dada por el método de aprendizaje a utilizar además de la topología usada, para que este aprendizaje sea óptimo es necesario contar con un buen algoritmo de aprendizaje y además disponer de los pares de entrenamiento adecuados.

2.4.2. Auto-organización.

Es posible manejar las imprecisiones e incertidumbre que aparecen al procesar información que conserva poco parecido con la disponible en su entrenamiento, así la RNA puede modificarse completamente con el objeto de llevar a cabo una tarea específica.

Biológicamente, un cerebro aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. De la misma manera, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de pesos, que son ajustados durante la fase de aprendizaje. En una RNA entrenada, el conjunto de los pesos determina el conocimiento de esa RNA y tiene la propiedad de resolver el problema para el que la RNA ha sido entrenada.

2.4.3. Tolerancia a Fallos

Una RNA posee una capacidad inherente a fallos, ya que la falla de un número no mayor de neuronas de la red no provoca una caída repentina de la red, aun cuando el comportamiento del sistema si se vea afectado. Esta tolerancia puede apreciarse en dos ocasiones, la primera de ellas, se da ante la capacidad de la red de reconocer patrones aun en la presencia de ruido o distorsiones, denominado tolerancia a fallos con respecto a los datos; la segunda se da cuando ocurre una pérdida de segmentos de la red, por lo que se pierden conexiones, aun así la red puede seguir funcionando.

2.4.4. Operación en tiempo real

Tiene relación con la capacidad de las RNA de realizar sus tareas en forma paralela, de esta forma las respuestas que se obtienen son el tiempo real, lo que aumenta el espectro de aplicaciones en las áreas se pueden utilizar este tipo de redes.

Estas son algunas de las ventajas características más importantes del cerebro, otras son la flexibilidad, la capacidad de aprender con rapidez y generalizar, la alta velocidad de procesamiento y el buen manejo de información ruidosa e inconsistente. Todo esto ha hecho que las RNA's sean consideradas como una herramienta computacional multidisciplinaria, centrada principalmente en áreas como las matemática, estadística, física, neurociencia, ciencias de la computación e ingeniería entre otras.

2.5. Topologías de redes neuronales

La topología o arquitectura de la red neuronal consiste básicamente en cómo se organizan y se disponen las neuronas dentro de la misma, formando agrupaciones de neuronas que son conocidas como capas, definiendo además cual es la conexión entre estas capas desde la capa de entrada, capas intermedias y capa de salida. Debido a esto se tiene que los parámetros principales de una red neuronal son el número de capas de esta, el número de neuronas que participa en cada capa, la interconectividad de estas y el tipo de conexión que se establece entre estas. A continuación se describirán topologías de redes neuronales y se establece la diferencia entre estas.

2.5.1. Redes de Propagación hacia Adelante (Feed-Forward)

En la Red la señal de entrada es propagada en solo un sentido, hacia adelante, así los nodos de una capa sólo se conectan con los nodos de capas posteriores, sin generar ciclos de información. También reciben el nombre de redes estáticas, debido a que el estado interno de

la red no depende de estados previos de la misma, producto de la falta de ciclos de propagación.

Las redes feed-forward más conocidas son: *Perceptron*, *Adaline*, *Madaline*, *Linear Adaptive Memory*, *Drive-reinforcement*, *Back-propagation* [9]. Todas ellas son útiles en aplicaciones de reconocimiento o clasificación de patrones.

2.5.1.1. Perceptron Multicapa

Se desprende la arquitectura Feed-forward siendo uno de los modelos más difundidos dentro de dicho grupo, debido fundamentalmente a la cualidad de aproximador universal, ya que puede aproximar cualquier función continua sobre un compacto de \mathbb{R}^n , debido a que una gran cantidad de neuronas puede activarse con un solo patrón de entrada, este tipo de arquitectura posee al menos una capa oculta.

Las funciones de activación de un Perceptron [10] pueden ser sigmoidea, tangente hiperbólica o la función escalón (*funciones con forma de S*). Dicha propiedad tiene la desventaja de que el aprendizaje de un Perceptron Multicapa sea lento, dado que un cambio en un solo peso de la red provoca cambios que se traspasan a una gran cantidad de neuronas, reduciendo el efecto de ciclos previos de entrenamiento.

El entrenamiento de un Perceptron Multicapa es usualmente desarrollado mediante el algoritmo Back-propagation, uno de los algoritmos con más importancia histórica en el desarrollo de las redes neuronales.

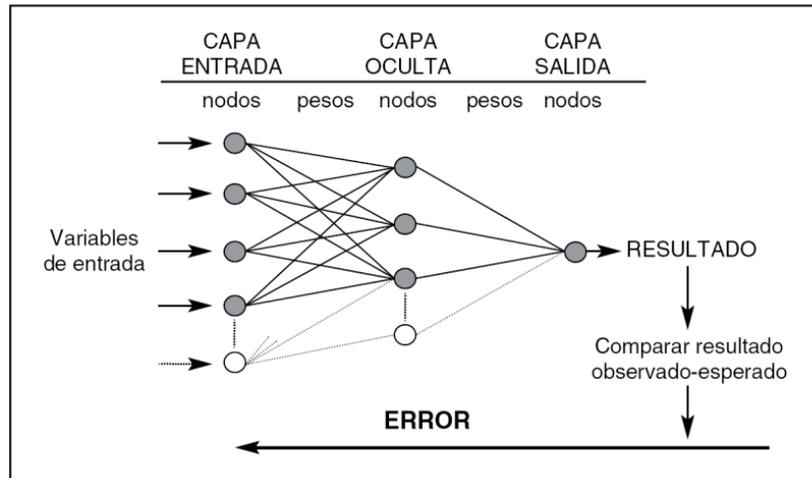


FIGURA 2-3 : RED PERCEPTRON MULTICAPA

2.5.1.2. Red de Base radial

Una red neuronal artificial de Base Radial, en adelante red RBF (Radial Basis Function), muestra una arquitectura bastante similar a un Perceptron Multicapa de 2 capas, en el sentido que es una red feed-forward con una capa de entrada que propaga la información a la primera y única capa oculta. Además, en la capa de salida se utiliza una función de activación lineal, por lo que podemos decir que la salida de una red RBF es una combinación lineal de

transformaciones no lineales de los patrones de entrada. La razón de que se utilice una transformación no lineal, seguida de una transformación lineal se justifica matemáticamente mediante el teorema de Cover [8] sobre la separabilidad de patrones:

“Un problema de clasificación de patrones transformado no linealmente a un espacio de dimensión superior tiene mayor probabilidad de ser linealmente separable que en un espacio de dimensión menor”.

Tomando en cuenta el teorema anterior, se concluye que al aumentar la cantidad de nodos en la capa oculta de una red RBF, aumentaremos la probabilidad de separabilidad lineal, demostrando la gran utilidad de las redes RBF en problemas de clasificación.

Pese a que las redes RBF junto con la del Perceptron Multicapa son consideradas aproximadores universales, estas poseen diferencias estructurales, dentro de las cuales se encuentran:

1. El Perceptron Multicapa puede poseer más de una capa oculta, mientras que una red RBF normalmente posee sólo una capa oculta.
2. Las redes RBF no poseen pesos asociados entre la capa de entrada y la capa oculta. En vez de eso, los parámetros de la capa oculta de la red son los centros y la anchura de la función de base radial.
3. La capa de salida de una red RBF posee funciones de activación lineales, lo cual no es imprescindible en un Perceptron Multicapa.
4. La función de activación de la capa oculta de una red RBF es una función de base radial, lo cual determina el carácter local de las aproximaciones construidas por la red; frente a las funciones de activación sigmoidea de los Perceptrones Multicapa, que construyen aproximaciones globales.

De la lista anterior, los puntos 2 y 4 son claves para diferenciarlas de los Perceptrones Multicapa (PM) y forman la base de la arquitectura de una red RBF.

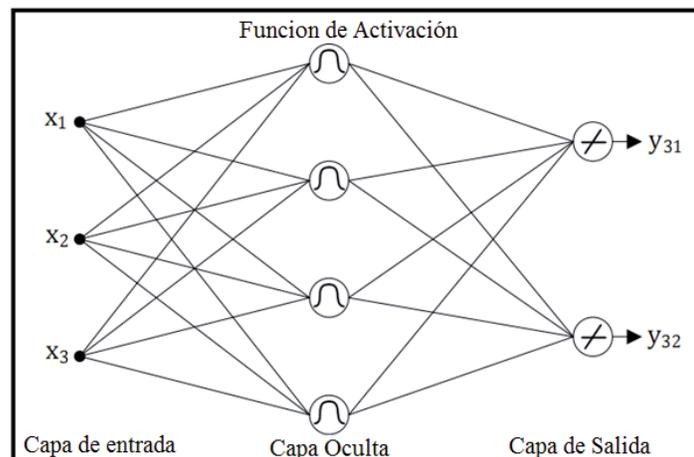


FIGURA 2-4 : ESTRUCTURA DE UNA RED RBF.

2.5.2. Redes de Retro-Propagación (Feed-Back)

Son redes donde los nodos de una capa se conectan con nodos de capas posteriores y anteriores generando ciclos de propagación. Los ciclos de propagación pueden generarse en tres niveles: entre capas, donde una neurona se conecta con otras de distinta capa; Intercapa, donde la neurona se conecta con neuronas de su misma capa y finalmente intra-neurona, donde la neurona toma como entrada la salida de la misma. También reciben el nombre de redes dinámicas, dado que el estado interno de la red depende de un estado previo de la misma, en una red bicapa, existen dos conjuntos de pesos, los correspondientes a la primera capa (capa de entrada) feed-forward y los pesos de la segunda capa feed-back (capa de salida). Los valores de los pesos de estos tipos de conexiones no tienen porqué coincidir, siendo diferentes en la mayor parte de los casos.

Este tipo de estructura (bicapa) es particularmente adecuada para realizar una asociación de una información o patrón de entrada (en la primera capa) con otra información o patrón de salida en la segunda capa (lo cual se conoce como heteroasociación). A continuación se analizaran dos tipos de redes recurrentes basadas en Perceptron Multicapa.

2.5.2.1. RMLP Jordan

La primera arquitectura recurrente derivada a partir de una red feed-forward fue planteada por Jordan en 1986 (*M. I. Jordan, 1990*). En la figura a continuación, la red de Jordan presenta una topología de capas similar a las redes feed-forward, con la inclusión de una capa de contexto que realimenta las señales de salida hacia las capas escondidas.

En una red de Jordan, existen retardos en todas las realimentaciones. Adicionalmente, se considera que los pesos de las realimentaciones desde la capa de salida hacia la capa de contexto pertenecen a un vector unitario [11] - [12].

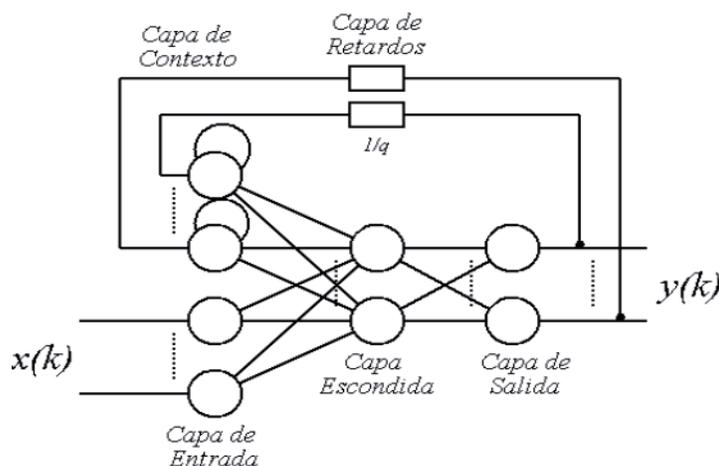


FIGURA 2-5 : RED NEURONAL DE JORDAN

En esta red las conexiones recurrentes tienen un parámetro asociado μ (generalmente positivo y menor que 1), las neuronas de contexto reciben una copia de las neuronas de salida y de ellas mismas.

La activación de las neuronas de contexto en t:

$$c_i(t) = \mu c_i(t-1) + y_i(t-1) \text{ para } i = 1, 2 \dots m \quad (2.3)$$

La entrada total de la red es una concatenación de las activaciones de las neuronas de entrada y de las neuronas de contexto.

$$u(t) = (x_1(t), \dots, x_n(t), c_1(t), \dots, c_m(t)) \quad (2.4)$$

Neuronas ocultas y de salida generalmente usan una función de activación sigmoidea, las neuronas de contexto usan una función de activación lineal, así las neuronas de contexto en desarrollada en t está dada por:

$$\begin{aligned} c_i(t) &= \mu c_i(t-1) + y_i(t-1) \text{ para } i = 1, 2 \dots m & (2.5) \\ c_i(t-1) &= \mu c_i(t-2) + y_i(t-2) \\ c_i(t) &= \mu^2 c_i(t-2) + \mu y_i(t-2) + y_i(t-1) = \dots = \\ &= \mu^{t-2} y_i(1) + \mu^{t-3} y_i(2) + \dots + \mu y_i(t-2) + y_i(t-1) \end{aligned}$$

- Las neuronas de contexto acumulan las salidas de la red para todos los instantes anteriores
- El valor de μ determina la sensibilidad de estas neuronas para retener dicha información
- μ próximo a 0 \rightarrow estados alejados en el tiempo se olvidan con facilidad
- μ próximo a 1 \rightarrow estados alejados en el tiempo se memorizan con facilidad

2.5.2.2. RMLP Elman

En 1990, Elman modifica esta arquitectura, considerando realimentaciones desde capas escondidas hacia la capa de contexto, sin considerar realimentaciones locales [12], [13]. En la figura 2-13 se muestra una red de Elman, también denominada red recurrente simple [13].

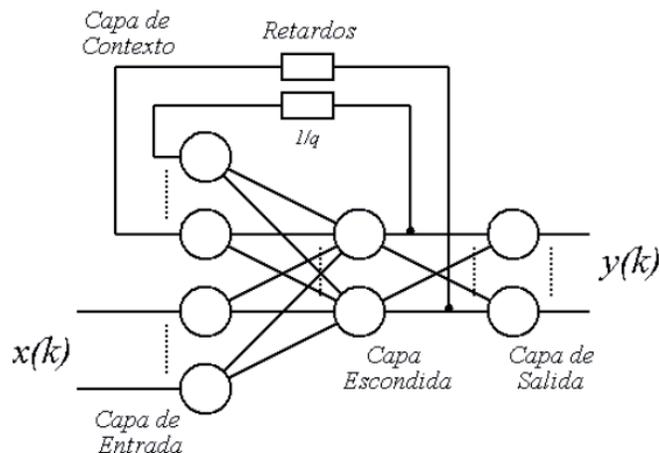


FIGURA 2-6 : RED NEURONAL DE ELMAN

La red de Elman generalmente posee neuronas con función transferencia sigmoidea en su capa oculta, es el caso *tansig* y neuronas con función de transferencia tipo lineal en la capa de salida, está el caso *Purelin*, la ventaja de la configuración de esta red de dos capas con este tipo de funciones de transferencia, es que según lo demostrado por Funahashi [14], puede aproximar cualquier función con la precisión deseada mientras que esta posea un número finito de discontinuidades, para lo cual la precisión de la aproximación depende de la selección del número adecuado de neuronas en la capa oculta.

Para la red de Elman la capa oculta es la capa recurrente y el retardo en la conexión de realimentación almacena los valores de la iteración previa, los cuales serán usados en la siguiente iteración; Dos redes de Elman con los mismos parámetros y entradas idénticas en las mismas iteraciones podrían producir salidas diferentes debido a que pueden presentar diferentes estados de realimentación.

La red de Elman tiene tantas neuronas de contexto como ocultas, en esta no existe un parámetro asociado a la conexión recurrente, la función de activación de las neuronas de contexto viene dada por:

$$c_i(t) = a_i(t - 1) \text{ para } i = 1, 2, \dots, r \quad (2.6)$$

El resto de las activaciones se calculan como en una red feed-forward considerando como entrada el vector total:

$$u(t) = (x_1(t), \dots, x_n(t), c_1(t), \dots, c_m(t)) \quad (2.7)$$

2.5.3. Funciones de activación

2.5.3.1. Funciones de activación Sigmoides

La función sigmoidea es el tipo de función de activación usada con mayor frecuencia para las aplicaciones de las RNA. Su nombre se debe a la forma de S que presenta la gráfica que se apreciará a continuación, y está definida por una función estrictamente creciente, no lineal y diferenciable, condiciones deseables para el proceso de entrenamiento de la red. Además la condición de diferenciable, es necesaria para que se pueda aplicar la rutina de entrenamiento basada en las arquitecturas de feed-back, base radial, etc.

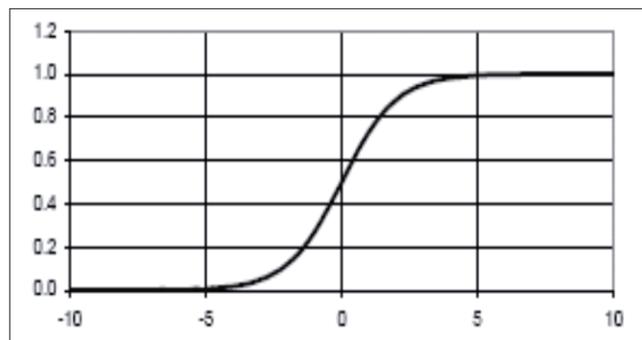


FIGURA 2-7 : FUNCIÓN DE ACTIVACIÓN SIGMOIDAL.

Por otra parte, las funciones sigmoideas de mayor uso en la literatura son la función sigmoidea logarítmica, representada en la figura anterior, definida matemáticamente por $\varphi(\cdot) = 1/(1 + e^{-a \cdot v})$, donde “a” representa la pendiente de la curva; y la función sigmoidea tangencial, definida matemáticamente por $\varphi(v) = a \operatorname{Tanh}(bv)$, donde “a” y “b” son constantes positivas. Aunque estas dos funciones sólo difieren en la escala y en la simetría alrededor de cero, se ha comprobado experimentalmente que la sigmoidea tangencial ofrece una mejor estabilidad numérica en el entrenamiento de la RNA.

Los valores de salida de esta función están comprendidos entre 0 y 1. El modificar el valor de “a” tiene influencia en la pendiente de la curva.

2.5.3.2. Wavelet

Como se vio en el punto anterior, la función de activación es uno de los parámetros más importantes dentro del modelo de la neurona, ya que define la salida de una neurona en términos de su potencial de activación, en este trabajo la función de activación será la transformada Wavelet (TW), en este apartado se dará a conocer cuáles son sus aplicaciones, su clasificación y las ventajas que conlleva su uso.

Las Wavelet pertenecen a la familia de funciones del tipo:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right); a, b \in \mathbb{R}, a \neq 0. \quad (2.8)$$

Las Wavelet son generadas a partir de una Wavelet madre, en la cual se aplican dos variables, la primera es llamada variable de escala “a” cuya función es la de permitir dilataciones y contracciones de la señal, y la segunda es la variable de traslación “b”, que permite mover la variable en el tiempo.

Existen diferentes wavelets que son ya utilizadas de forma constante y que tienen definiciones establecidas, tales como los parámetros de la función. Sin embargo la elección de un tipo u otro de wavelet depende de la aplicación específica que se le vaya a dar, una correcta elección, nos da una mejor aproximación a la realidad.

Sobre las wavelets hay que considerar que cumplen con ciertas características, una de estas es el momento de desvanecimiento, este concepto es muy importante, ya que nos permite conocer la forma de la wavelet y es un parámetro para saber qué tan hábil es la wavelet para suprimir un polinomio dado. La suavidad de la wavelet está limitada por el número de momentos de desvanecimiento que tenga. El i-ésimo momento de la calculan con la siguiente integral:

$$\int_{-\infty}^{\infty} \psi(x) \psi^i dx = 0 \quad (2.9)$$

Existen otras características importantes de las wavelets tales como el soporte compacto, que es la propiedad de que la wavelet sea de duración finita, lo que permite una menor complejidad en los cálculos, mejor resolución en tiempo y pobre resolución en frecuencia. También está la propiedad de simetría que permite que los filtros sean de fase lineal. La

ortogonalidad es la propiedad que se logra cuando el producto punto de dos vectores es igual a cero y es importante en los estudios encontrar este tipo de características para que los análisis sean estables.

Funciones Wavelets más usadas

En la familia de las wavelets, las que son constantemente usadas y que cuentan con sus parámetros definidos son cuatro, las que se verán a continuación.

Wavelet Haar

Dentro de las wavelets más famosas y utilizadas podemos citar la Haar, esta es la más simple y la más antigua de las wavelet, se describe con la siguiente función:

$$\psi(x) = \begin{cases} 1; & 0 \leq x \leq 1/2 \\ -1; & 1/2 \leq x \leq 1 \\ 0; & \text{Otro valor} \end{cases} \quad (2.10)$$

Su gráfica se muestra en la figura 6, en esta podemos ver que efectivamente es una wavelet sencilla. Sin embargo es muy utilizada para análisis de señales usando transformadas discretas y continuas. Tiene un sólo momento de desvanecimiento, el concepto de desvanecimiento viene dado por el promedio de la wavelet, cuando el valor promedio es cero se tiene un momento de desvanecimiento.

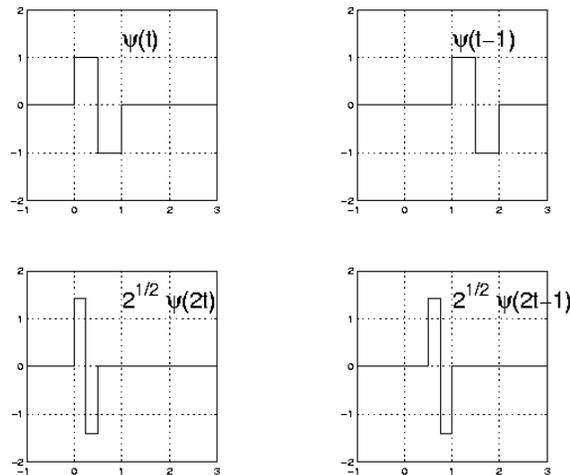


FIGURA 2-8 : WAVELET HAAR.

Wavelet Mexican Hat

Esta Wavelet también es usada en el análisis de señales, recibe su nombre ya que su grafica es semejante en forma a la de un sombrero mexicano, la función matemática que representa a esta wavelet está dada por:

$$mex \psi(x) = \frac{2(1 - x^2)e^{-x^2/2}}{\pi^{1/4} \sqrt{3}} \quad (2.11)$$

Esta es la segunda derivada de la función de densidad de probabilidad wavelet gaussiana. Como vemos en la figura 2-9 es la forma de la wavelet es simétrica, característica que le permite examinar a las señales de un modo simétrico y lineal en la fase, igual que la wavelet Haar. No todas las wavelets son simétricas, pero también se utilizan en los análisis dependiendo del objetivo que se quiera lograr con ellas.

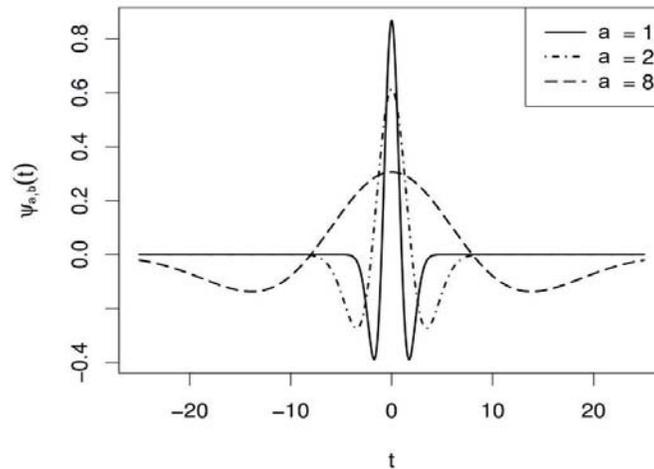


FIGURA 2-9 : WAVELET MEXICAN HAT.

Wavelet Daubechies

Esta Wavelet puede llegar a ser de orden N dependiendo del número de momentos de desvanecimiento que se deseen, donde N es un entero positivo y denota el número de coeficientes de filtro que tiene esa wavelet, esta se caracteriza ya que existe una relación del doble entre los números de coeficientes de filtro y los momentos de desvanecimiento; por ejemplo la wavelet Daubechies de orden 1 es la wavelet Haar que hemos visto anteriormente, esta wavelet cuenta con las características de ortogonalidad y biortogonalidad, además de que se pueden realizar las transformadas de wavelets discretas y continuas con ella.

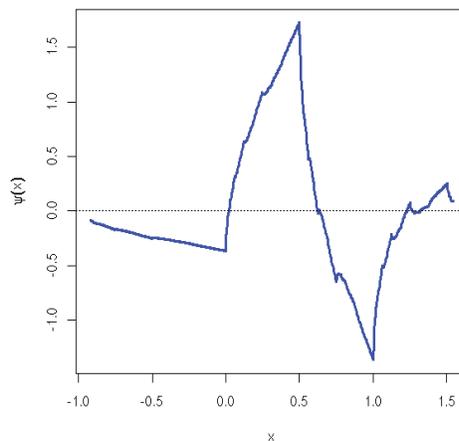


FIGURA 2-10 : WAVELET DAUBECHIES DE SEGUNDO ORDEN

Wavelet Morlet

En la figura 2-11 podemos ver que la wavelet Morlet es simétrica. Por otro lado debemos comentar que no posee características de ortogonalidad ni biortogonalidad, además de que sólo es útil para realizar la Transformada Continua de Wavelets. La grafica de la Wavelet Morlet es mostrada a continuación.

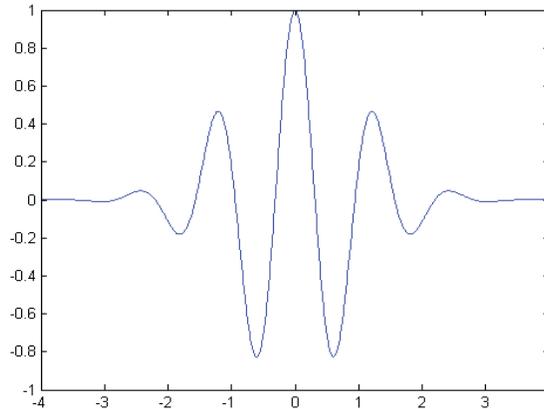


FIGURA 2-11 : WAVELET MORLET

Su expresión matemática viene dada por:

$$morl(x) = e^{-\frac{x^2}{2}} \cos(5x) \quad (2.12)$$

Wavelet Gaussiana

La Wavelet Gaussiana (WG) es otra muestra muy común de estas funciones finitas. Esta se define como la derivada de la función de densidad de probabilidad Gaussiana. Matemáticamente esta definición se aprecia como:

$$gaus(x, n) = Cn \frac{d}{dx} (e^{-x^2}, n) \quad (2.13)$$

Donde Cn es una constante determinada por $2 - norm de Gauss(x, n) = 1$.

Con esta wavelet sólo es posible realizar la transformada continua de wavelets y tiene la propiedad de ser simétrica o asimétrica dependiendo del valor de n .

En la siguiente, se presenta la wavelet Gaussiana con el valor de $n = 1$.

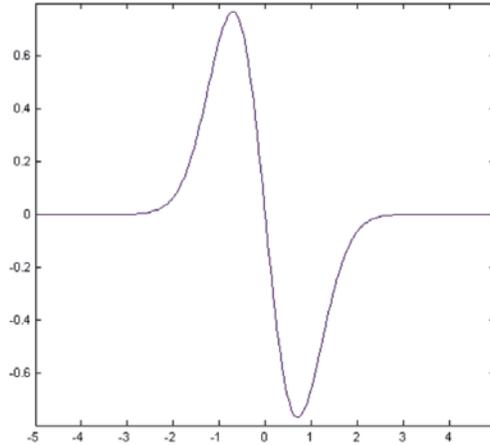


FIGURA 2-12 : WAVELET GAUSSIANA

Existen muchas otras wavelets en la literatura entre ellas podemos encontrar wavelet *Meyer*, *Spline*, *Biorthogonal*, *Symmlet*, y *Coiflet*, esto debido a la variedad de aplicaciones que se tiene al usar este tipo de funciones. En estos últimos años esto se ha reflejado en un explosivo desarrollo de aplicaciones en el campo de la investigación.

Transformadas Wavelets

Para poder definir distintas transformadas, se necesita definir en primera instancia cual será la wavelet madre a utilizar, ecuación 2.9, recordando que a , es el factor de escala y b el factor de corrimiento o traslación.

Una vez definida la wavelet madre, se considerara que la transformada wavelet de una $f(t)$ será la descomposición de la misma en un conjunto de funciones $\psi_{a,b}(t)$, estas forman una base y son consideradas como wavelets. Está definida como:

$$W_f(a, b) = \int f(t) \bar{\psi}_{a,b}(t) dt \quad (2.14)$$

Una vez definido esto podremos decir que existen dos tipos de transformadas, la discreta y la continua.

Transformada Continúa Wavelet

La Transformada Continua wavelet (CWT), de una señal está definida por las $f(t)$ Wavelets descritas en la siguiente ecuación 2.15.

$$W_f(a, b) = \frac{1}{\sqrt{|a|}} \int f(t) \bar{\psi}_{a,b}(t) dt; a, b \in \mathbb{R}, a \neq 0. \quad (2.15)$$

Donde “ a ” es la escala y “ b ” es la traslación, ambos números reales y $f(t)$ es la función a analizar. Por supuesto que si la escala es cero la wavelet no existirá y el análisis se indeterminara. Para cualquier otro valor de la escala, esta será una CWT.

Transformada discreta Wavelet

Para explicar cómo se realiza la transformación de una señal discreta partamos del hecho de que contamos con un par de escalas que cumplen con la condición que $a_0 < a_1$ corresponden aproximadamente a dos frecuencias $f_0 < f_1$, una forma natural de discretizar los parámetros de tiempo y escala es submuestrear de acuerdo a la regla de Nyquist, los coeficientes en escala a_1 en $\left(\frac{f_0}{f_1}\right)^{i-\text{ésimo}}$ a la razón de los coeficientes en la escala a_0 . Para Wavelets discretas los parámetros de escala y traslación son elegidos de tal manera que en el nivel j , la Wavelet $a_0^j \psi(a_0^{-j} x)$ es a_0^j veces el ancho de $\psi(x)$. Esto significa que el parámetro de escala es $a = a_0^j$; $j \in \mathbb{Z}$ y el parámetro de traslación $b = k b_0 a_0^j$; $j, k \in \mathbb{Z}$ [15]. De este modo la familia de wavelets está dada por:

$$\psi_{j,k}(x) = a_0^{-\frac{j}{2}} \psi(a_0^{-j} x - k b_0); j, k \in \mathbb{Z} \quad (2.16)$$

La Transformada Discreta de Wavelets (DTW) tiene la forma:

$$d_{j,k}(x) = a_0^{-\frac{j}{2}} \int f(x) \psi(a_0^{-j} x - k b_0) dx \quad (2.17)$$

Para recuperar $f(x)$ de los coeficientes $\{d_{j,k}\}$ debe existir la siguiente condición de estabilidad:

$$A \|f(x)\|^2 \leq \frac{2}{A+B} \sum \sum |d_{j,k}|^2 \leq B \|f(x)\|^2 \quad (2.18)$$

Con $A > 0$ y $B < \infty$ para todas las señales $f(x)$ en $L^2(\mathbb{R})$. Entonces la fórmula de reconstrucción está determinada por:

$$f(x) \approx \frac{2}{A+B} \sum \sum d_{j,k} \psi_{j,k}(x) \quad (2.19)$$

Entre más cercanos sean A y B más aproximada será la reconstrucción. Cuando $A = B = 1$ la familia de wavelets es ortonormal [15].

De lo anterior surge un concepto importante que es el Análisis de Multiresolución (MRA), que como lo dice el nombre es un análisis de la señal a estudiar de tal modo que cada componente de frecuencia es analizado con una resolución diferente. Esto es una alternativa más sobre la transformada de Fourier (TF) que analiza todas las componentes de frecuencia a una misma resolución dada. En general, las ventajas que ofrece este método es que, con el uso de wavelets a altas frecuencias se tiene una buena resolución en tiempo y poca resolución en frecuencia, mientras que para bajas frecuencias los resultados son buena resolución en frecuencia y poca resolución en tiempo. Para el caso de la transformación discreta debemos

tomar en cuenta un muestreo que convierta la señal continua en discreta y de ahí se trabaja de forma similar.

Aplicaciones de la transformada Wavelet

Los procesos antes señalados principalmente se usan en el procesamiento de datos y análisis de señales. En este apartado se vera cual es la aplicación que se le puede dar a la TW para usos en la vida real, acercandola y haciendo que sea mas familiar para el lector.

La Teoría de Wavelets tiene muchas aplicaciones reales que comprenden la detección de discontinuidades y puntos de ruptura en las señales, la identificación de frecuencias puras, la reducción de ruido en señales, la compresión de señales, aproximación de funciones, métodos espectrales para resolver ecuaciones diferenciales, análisis de fluidos turbulentos, entre otros. La que más nos interesa estudiar es, lógicamente, aquella aplicación con la que es posible reducir el ruido de las señales de interés. Sin embargo, mencionaremos brevemente cómo es que se consiguen algunas otras aplicaciones de las que se han mencionado anteriormente.

La que resulta muy fácil de entender a simple vista es la identificación de frecuencias puras, pues como hemos visto la Transformada de Wavelets (en sus 2 versiones) nos permite conocer las frecuencias que tienen la señal analizada, de este modo y con un tratamiento de extracción de la información adecuada para evitar que ésta información se corrompa es muy fácil identificar estas frecuencias puras.

La siguiente es la detección de discontinuidades y puntos de ruptura en las señales, es posible utilizar las wavelets para estos propósitos pues sus procesos de análisis de multiresolución nos permite conocer en que momento se presentan los cambios de frecuencias en la señal analizada, de éste modo podemos detectar algún punto en el que la señal tenga un cambio que puede ser un punto de ruptura o una discontinuidad.

La compresión de señales es, quizás, la aplicación más famosa de la Teoría de wavelets, dado que el FBI (Federal Bureau of Investigation) la ha utilizado para guardar las imágenes de las huellas digitales en sus archivos. Esta aplicación es posible gracias a los coeficientes de wavelet, información que sin ser la imagen misma posee los recursos suficientes, como hemos visto antes, para sintetizar la señal original, de este modo el total de información a guardar se reduce considerablemente, ya que no se guarda la imagen como tal, sino los coeficientes de wavelet que en comparación a la imagen original es notablemente menos información.

Después continúan otro tipo de aplicaciones que son propiamente los objetivos de estas aplicaciones y donde no se usan directamente los métodos aquí estudiados, pero si cooperan con la obtención del resultado final. Entre estas se encuentran las mencionadas anteriormente, métodos espectrales para resolver ecuaciones diferenciales y análisis de fluidos turbulentos. Es importante aclarar que no son todas las aplicaciones que existen para las wavelets, esto es sólo una prueba muy pequeña de lo que se puede conseguir aplicando las Transformadas o las wavelets por si solas. Tal y como lo marca la historia, el campo de aplicacion esta limitado por nuestra imaginación.

2.6. Aprendizaje de una red neuronal

El proceso de aprendizaje de una red neuronal implica el ajuste de todos los parámetros que intervienen en la red. En el caso de un Perceptron Multicapa o Monocapa el proceso consiste en ajustar los pesos de las conexiones entre las neuronas versus el ajuste de los centros y anchuras de las neuronas de la capa oculta. Para que se pueda realizar el aprendizaje, se debe contar con un parametro de comparacion de la solucion obtenida, esta esta fuertemente ligada al tipo de muestras que son utilizadas en esta etapa, es por esto que las muestras a utilizar deben ser significativas y representativas esto quiere decir que debe existir la cantidad suficiente de ejemplos y no varios ejemplos de el mismo tipo, ya que esto puede llevar a perder generalidad de la red y hacerla mas especifica.

Los metodos de aprendizaje son necesarios para poder calibrar la red, esta calibracion es posible mediante la introduccion paulatina de ejemplos a la red, proceso que se repite hasta alcanzar algun nivel de convergencia, el que indica cuando detener el proceso de entrenamiento. Esta convergencia puede definirse a traves de los siguientes parametros:

- **Verificación del error de los resultados:** Se define una función que determina que tan acertados son los resultados entregados por la red y se entrena hasta que la función de aceptación alcance un rango determinado.
- **Cuando el resultado del entrenamiento sea irrelevante:** Si se entrena a la red y se alcanza un punto en que los resultados entregados por esta son muy parecidos a los resultados entregados en la iteración anterior.
- **Alcanzando un número fijo de ciclos:** Los entrenadores de la red deciden un número fijo de iteraciones para entrenar a la red, una vez alcanzado dicho número, se detiene el entrenamiento independiente de la calidad de los resultados arrojado por la red.

Diferentes autores han propuesto modelos de aprendizaje, entre estos se encuentra el **aprendizaje supervisado**, y el **no supervisado**. El tipo de aprendizaje a utilizar en este trabajo será el aprendizaje supervisado, debido a que los datos de entrada serán previamente tratados y que se conoce el resultado deseado, a continuación se explicaran ambos y las características de estos.

2.6.1. Aprendizaje supervisado

El aprendizaje supervisado es aquel en el que el entrenador de redes neuronales, es decir, el diseñador, ha de indicar a la red tanto las entradas como las salidas que desea obtener, es decir, el diseñador ha de mostrar a la red las entradas y corregir sus salidas para que coincidan con unas salidas deseadas Para ello se regulan los pesos de la red, con el fin de obtener dicho resultado en un proceso que se denomina entrenamiento de la red, para esto se necesita un conjunto de datos de entrada previamente clasificado y definir cuál es el resultado objetivo. Ejemplos de este tipo de redes son: el Perceptron simple, la red Adaline, el Perceptron multicapa y la memoria asociativa bidireccional.

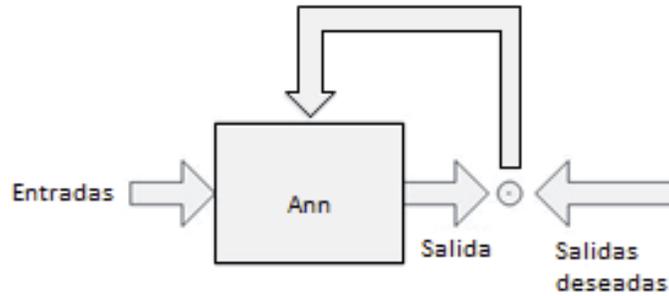


FIGURA 2-13 : APRENDIZAJE NEURONAL SUPERVISADO

Dentro de los algoritmos de aprendizaje supervisado podemos destacar al algoritmo Back-propagation, el cual es utilizado para el Perceptron Multicapa, este algoritmo toma un patrón de ejemplo aleatorio y lo pasa por la red, se obtiene una salida con la cual se le calcula el error cuadrático respecto del resultado esperado. Dado que la función de activación es una función no lineal, la función de error tampoco lo es, por lo tanto, se puede utilizar el método del descenso del gradiente, basado en la derivación de la función de activación, para obtener la dirección en la cual el error decrece. Con esta información podemos propagar el error obtenido por la capa de salida hacia las capas anteriores sucesivamente hasta llegar a la capa de entrada.

El proceso se repite hasta que el error total de la red (error promedio) tienda a 0 o hasta cumplir un cierto número de iteraciones. Es importante destacar que debido a la naturaleza del algoritmo, el cual está basado en la derivada, la red puede entregar resultados sub-óptimos al quedar atrapado en un mínimo local; lo cual puede solucionarse aumentando el número de neuronas ocultas, aumentando también el costo computacional asociado.

Se tiene también métodos como Levenberg-Marquardt [16], que corrige uno de los problemas del algoritmo Back-propagation, el cual consiste en encontrar una solución cualquiera, pudiendo quedar atrapada esta solución de máximo local y no de máximo global que es lo que se desea, otro método de obtención de los pesos y umbrales no dependiente de la derivada es el método de la matriz pseudo-inversa, el cual proporciona una solución directa y óptima al problema de la minimización del error aprovechando la linealidad de la capa de salida.

Existen diferentes formas de realizar el aprendizaje supervisado. Este tipo de aprendizaje necesita un profesor que mida el rendimiento del sistema.

- **Aprendizaje por corrección de error:** El entrenamiento consiste en presentar al sistema un conjunto de pares de datos, representando la salida y la entrada para dicha entrada. Este conjunto recibe el nombre de conjunto de entrenamiento. El objetivo consiste en minimizar el error entre la salida deseada y la actual es un aprendizaje off-line
- **Aprendizaje por refuerzo:** Es más lento que el anterior y no se conoce la salida exacta para cada entrada sólo se conoce como debería ser el comportamiento general ante diferentes entradas. En este caso la función del supervisor es más la de un crítico que la de un maestro. Es un aprendizaje on-line que produce una señal de refuerzo indicando el éxito o fracaso de la salida producida por la red. La red

actualiza sus pesos en función de la señal de refuerzo que dado una acción tomada por el sistema es seguida de un estado satisfactorio dicha acción es reforzada y disminuida en caso contrario.

- **Aprendizaje Estocástico:** Este aprendizaje consiste en realizar cambios aleatorios en los valores de los pesos y evaluar su efecto a partir del objetivo deseado y distribuciones de probabilidad.

2.6.2. Aprendizaje no supervisado

En el aprendizaje no supervisado no existe información referente a las salidas de los patrones que vamos introduciendo, sólo sus entradas y el estado interno de la misma. Ha de ser la red, por lo tanto, la que vaya adecuando sus pesos en función de la información interna que vaya recogiendo de las entradas.

Este tipo de entrenamiento se utiliza principalmente para clasificar y diferenciar rasgos significativos de un conjunto de datos no clasificado a priori, dado que la red internamente intenta encontrar redundancias y rasgos significativos para agrupar los datos, ejemplos de este tipo de redes son: las memorias asociativas, las redes de Hopfield, la máquina de Bolzman y la máquina de Cauchy, las redes de aprendizaje competitivo, las redes de Kohonen o mapas auto organizados y las redes de resonancia adaptativa (ART).

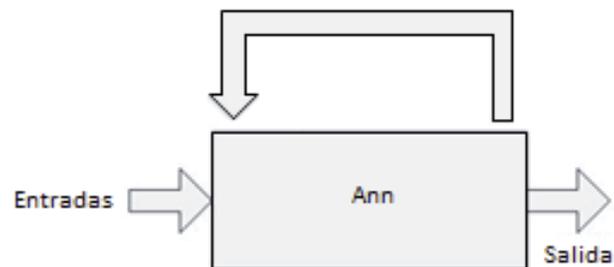


FIGURA 2-14 : APRENDIZAJE NEURONAL NO SUPERVISADO

3. Algoritmos Genéticos

3.1. Introducción

El ser humano en su afán de superación, ha buscado incansablemente la forma de poder predecir lo que pasara a su alrededor, para así poder controlar mejor sus acciones, sacando el máximo beneficio ejerciendo un máximo esfuerzo.

Los algoritmos genéticos (AG), fueron inventados en 1975 por John Holland, de la Universidad de Michigan. Los AG vistos de manera simplificada, algoritmos de optimización, es decir, tratan de encontrar la mejor solución a un problema dado entre un conjunto de soluciones posibles. Esta búsqueda de solución puede verse como el proceso de evolución biológica en sentido metafórico.

Es así como a fin de los ochenta, el interés de la comunidad científica por temas computacionales inspirados en la biología, ha visto como con su desarrollo les llevaba a cosas inimaginables, pasando de la implementación de redes neuronales, luego el aprendizaje de estas y por último el máximo exponente de los algoritmos genéticos, la computación evolutiva.

La inteligencia artificial es responsable de muchos de esos logros. Los pioneros de esta ciencia estaban tan interesados en la electrónica, como en la biología, es por esto que sus aplicaciones iban desde calcular trayectorias de misiles a tratar de modelar el comportamiento del cerebro tratando de imitar el proceso de aprendizaje humano y de simular la evolución biológica.

Los algoritmos genéticos se basan en el fenómeno de la evolución biológica, que implica una serie de cambios en el nivel más elemental de los organismos vivos, la célula. En biología, la información de un individuo está codificada en sus cromosomas, cadenas de ADN que se encuentran en cada una de las células de dicho individuo. Cada cromosoma puede dividirse en un cierto número de genes, los cuales codifican una característica particular del individuo que lo posee, la evolución puede interpretarse como una serie de combinaciones de cromosomas y la mutación de los mismos, estas combinaciones o cambios a su vez generan nuevos individuos que pueden ser más o menos aptos de sobrevivir en el entorno que les rodea, por lo que la evolución genera nuevos individuos que a través del tiempo se adaptan al medio que los rodea, sobreviviendo solo los más aptos.

La visión biológica sobre las especies da cuenta que estas han ido mutando, cambiando de forma, de manera de adaptarse y sobrevivir. Estos cambios, estas mutaciones y adaptaciones son las mismas que mediante AG, se dotara a una red neuronal de forma de hacer que esta se adapte.

3.2. Reseña Histórica

Los Algoritmos Genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acuerdo con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin en el origen de las especies. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

Es por esto que John Holland, de la Universidad de Michigan, desde pequeño y teniendo en cuenta la teoría evolutiva de Darwin, se preguntaba cómo logra la naturaleza, crear seres cada vez más perfectos, capaces de adaptarse completamente a su entorno todo esto dependiendo del concepto de perfección.

En los años 50, Holland entró en el mundo de la computación, pero fue a principios de los 60, en la Universidad de Michigan en donde sus ideas comenzaron a desarrollarse y a dar frutos. Inspirado por el biólogo R. A. Fisher con su libro “La teoría genética de la selección natural”, fue en donde comenzó a descubrir los medios de llevar a cabo sus propósitos de comprensión de la naturaleza.

Por tanto, cuando Holland se enfrentó a los algoritmos genéticos, los objetivos de su investigación fueron dos:

- Imitar los procesos adaptativos de los sistemas naturales,
- Diseñar sistemas artificiales (normalmente programas) que retengan los mecanismos importantes de los sistemas naturales.

Una de las mayores innovaciones propuestas por Holland fue la utilización de poblaciones de más de un individuo y la capacidad de cruzamiento de estos. Antes de la concepción de los algoritmos genéticos, las técnicas evolutivas utilizaban a sólo un individuo y se obtenía su descendencia a partir de mutación.

Unos 15 años más tarde, David Goldberg [17], estudiante de Holland, fue uno de los primeros que trató de aplicar los algoritmos genéticos a problemas industriales. Aunque Holland trató de disuadirle, porque pensaba que el problema era excesivamente complicado como para aplicar algoritmos genéticos, Goldberg consiguió lo que quería, escribiendo un algoritmo genético en un ordenador personal Apple II. Estas y otras aplicaciones creadas por estudiantes de Holland [18] convirtieron a los algoritmos genéticos en un campo con base suficiente aceptado para celebrar la primera conferencia en 1985.

3.3. Operadores de un Algoritmo Genético

Los elementos de un algoritmo genético vienen dados por su representación biológica, así podemos decir que todos los individuos están compuestos por una o más células, cada una de las cuales contiene a su vez uno o más cromosomas, entendiéndose cromosomas como

cadena de ADN, que tienen la función de ser una especie de “anteproyecto” del organismo del que forman parte.

3.3.1. Cromosomas

Un cromosoma se puede dividir, conceptualmente, en genes, bloques funcionales de ADN que codifican una determinada proteína. Los genes son representados en un vector, estos son los responsables de determinar los rasgos del individuo. Las diferentes posibilidades de escoger un rasgo específico tal como el color de ojos reciben el nombre de alelos. Los individuos resultantes, tanto como sus padres son considerados como posible solución al problema a resolver.

3.3.2. Presión selectiva del medio

Dentro del espacio de soluciones, existen soluciones que no son factibles, la interpretación biológica de esta frase daría cuenta que en un espacio determinado, no existiría espacio para todos, ni recursos suficientes para sustentarlos a todos por lo que de entre los individuos resultantes, los que menos se adaptan a esta situación tienen menos probabilidades de sobrevivir y de aparearse.

3.3.3. Cruce de Cromosomas

Para poder realizar el cruce se necesitan de dos individuos, llamados padres, de estos se pueden obtener nuevos individuos, llamados descendientes o hijos, el cruce se realiza mediante un operador que actúa sobre los cromosomas de los individuos, este es el encargado de elegir un lugar de cruce dentro del cromosoma, cambiar las secuencias antes y después de esa posición para crear nueva descendencia. Se busca imitar la recombinación biológica entre dos organismos haploides, el resultado del cruce pretende ofrecer soluciones potencialmente mejores que las iniciales (padres).

3.3.4. Mutación de cromosomas

Este operador produce variaciones de modo aleatorio en un cromosoma. La mutación puede darse en cada posición de un bit en una cadena (alelo), con una probabilidad, normalmente muy pequeña, aumentando la probabilidad de no exista lugar del espacio de búsqueda sin visitar.

Para poder llegar a la solución deseada de un problema en particular se deben tener en cuenta todos los factores nombrados con anterioridad, así en primera instancia las soluciones factibles se generan de manera aleatoria, mediante la generación de descendientes se busca obtener un conjunto de soluciones, de las cuales se determinan óptimas, en la naturaleza lo primordial es optimizar la supervivencia, eso significa a su vez maximizar diversos factores y minimizar otros. Un algoritmo genético, sin embargo, se usará habitualmente para optimizar sólo una función, no diversas funciones relacionadas entre sí simultáneamente. Mediante el cruzamiento y la mutación, los individuos generados permiten explorar el espacio de soluciones, generando en cada etapa individuos que están más cercanos a la solución global deseada.

3.4. Algoritmo Genético Simple

Este algoritmo llamado también algoritmo genético básico, o canónico, parte considerando los elementos de un algoritmo genético y de acuerdo con estos se generan etapas descritas a continuación.

Sea X el problema a resolver. Dada una representación de candidatas a soluciones en una cadena de bits, un algoritmo genético simple, tal y como se describe en Mitchell M. (1998), trabajaría del siguiente modo:

1. Comenzar con una población P generada aleatoriamente de n cromosomas de t bits.
2. Calcular la capacidad $f(x)$ para cada cromosoma x de P .
3. Repetir los siguientes pasos hasta que se hayan creado n descendientes:
 - Seleccionar un par de cromosomas padre de P , siendo la probabilidad de selección una función creciente de la capacidad. La selección se realiza “con remplazo”, es decir, que el mismo cromosoma puede ser seleccionado en más de una ocasión para ser padre.
 - Con probabilidad P_c (probabilidad de cruce, o tasa de cruce), cruzar el par en un punto elegido aleatoriamente (con probabilidad uniforme) para formar dos descendientes. Si no tiene lugar ningún cruce, formar dos descendientes que sean copias exactas de sus respectivos padres. (Obsérvese que aquí la probabilidad de cruce se define como la probabilidad de que dos padres se crucen sobre un único punto. Hay otras versiones de algoritmos genéticos que son de “cruces en múltiples puntos”, en los que la tasa de cruce para una pareja de padres es el n° de puntos en los que tiene lugar un cruce).
 - Mutar los dos descendientes en cada lugar con probabilidad p_m (probabilidad de mutación, o tasa de mutación), y colocar los cromosomas resultantes en la nueva población P' . Si n es impar, se puede rechazar aleatoriamente a un miembro de la nueva población.

4. Reemplazar la población actual P con la nueva P' .

5. Volver al paso 2.

Al traspasar este algoritmo al pseudo lenguaje se tiene:

```
BEGIN /* Algoritmo Genético Simple */
```

```
    Generar una población inicial.
```

```
    Computar la función de evaluación de cada individuo.
```

```
WHILE NOT Terminado DO
```

```
    BEGIN /* Producir nueva generación */
```

FOR Tamaño población/2 DO

BEGIN /*Ciclo Reproductivo */

- **Seleccionar** dos individuos de la anterior generación, para el cruce (probabilidad de selección proporcional a la función de evaluación del individuo).

- **Cruzar** con cierta probabilidad los dos individuos obteniendo dos descendientes.

- **Mutar** los dos descendientes con cierta probabilidad.

- Computar la función de evaluación de los dos descendientes mutados.

- **Insertar** los dos descendientes mutados en la nueva generación.

END

IF la población ha convergido **THEN**

Terminado:= **TRUE**

END

END

3.5. Codificación del cromosoma

Existen diferentes tipos de codificación que se puede usar para el tratamiento del cromosoma, tales como La codificación binaria, la codificación real, alfabética, y la codificación de árbol, el proceso que describen las diferentes tipos de codificación es análogo y será descrito a continuación.

Los individuos de una población pueden representarse como un conjunto de parámetros, denominados genes, los cuales agrupados forman una ristra de valores denominados cromosomas.

Generalmente se utiliza una codificación de los individuos basados en el alfabeto $\{0,1\}$, esto no implica que no se pueda ocupar otro tipo de representación.

En términos biológicos, el conjunto de parámetros representando un cromosoma particular se denomina *fenotipo*. El fenotipo contiene la información requerida para construir un organismo, el cual se refiere como *genotipo*. Los mismos términos se utilizan en el campo de los Algoritmos Genéticos. La adaptación al problema de un individuo depende de la evaluación del *genotipo*. Esta última puede inferirse a partir del *fenotipo*, es decir puede ser computada a partir del cromosoma, usando la función de evaluación.

La *función de evaluación* debe ser diseñada para cada problema de manera específica. Dado un cromosoma particular, la función de adaptación le asigna un número real, que se

supone refleja el nivel de adaptación al problema del individuo representado por el cromosoma.

Durante la *fase reproductiva* se seleccionan los individuos de la población para el cruzamiento y producir descendientes, que constituirán una vez mutados la siguiente generación de individuos. La selección de padres se efectúa al azar usando un procedimiento que favorezca a los individuos mejor adaptados ya que a cada individuo se le asigna una probabilidad de ser seleccionado que es proporcional a su función de adaptación. Este procedimiento se dice que está basado en la ruleta sesgada, en donde los individuos mejor evaluados tienen la opción de ser elegidos más veces para realizar la reproducción.

Una vez seleccionados dos padres, sus cromosomas se combinan, utilizando habitualmente los operadores de cruce y mutación, para esto el operador de cruce, coge dos padres seleccionados y corta sus ristas de cromosomas en una posición escogida al azar, para producir dos subristras iniciales y dos subristras finales, luego se intercambian las subristras finales, produciéndose dos nuevos cromosomas completos. Ambos descendientes heredan genes de cada uno de los padres. Este operador se conoce como operador de cruce basado en un punto. Habitualmente el operador de cruce no se aplica a todos los pares de individuos que han sido seleccionados para emparejarse, sino que se aplica de manera aleatoria, normalmente con una probabilidad comprendida entre 0.5 y 1.0. En el caso en que el operador de cruce no se aplique, la descendencia se obtiene simplemente duplicando los padres.

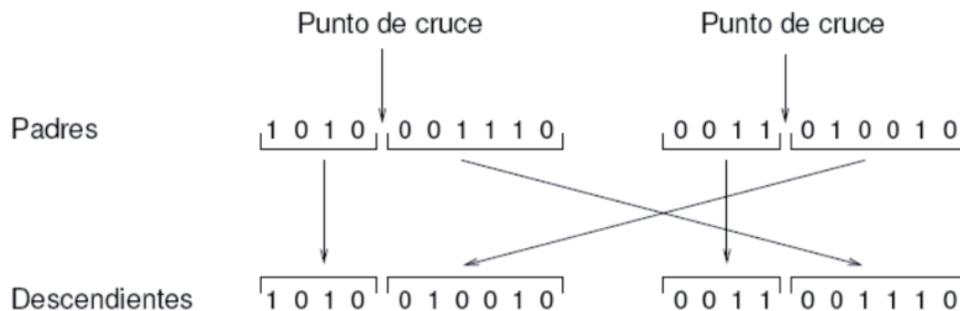


FIGURA 3-1 : OPERADOR DE CRUCE

El operador de mutación se aplica a cada hijo de manera individual y consiste en la alteración aleatoria (normalmente con probabilidad pequeña) de cada gen componente del cromosoma. Si bien puede en principio pensarse que el operador de cruce es más importante que el operador de mutación ya que proporciona una exploración rápida del espacio de búsqueda, este último asegura que ningún punto del espacio de búsqueda tenga probabilidad cero de ser examinado y es de capital importancia para asegurar la convergencia de los Algoritmos Genéticos. Este proceso es análogo para todo tipo de codificación, a continuación se explicara las diferencias entre estos.

3.5.1. Codificación Binaria

En el punto anterior se habló del alfabeto $\{0,1\}$, este alfabeto será usado para este tipo de codificación, debido a que un cromosoma binario es una cadena de bits 0 y 1, un ejemplo cromosoma es visible en la figura 3-1. En dicho ejemplo cada uno de los genes puede tomar el

alelo “0” o el alelo “1”, y cada posición del arreglo de números binarios representa un gen con un cierto significado. Las primeras investigaciones en genéticos utilizaron este tipo de codificación debido a su sencillez, además siendo el primer tipo de codificación propuesta.

Una de las desventajas que posee este tipo de codificación es el largo que puede llegar a tener un cromosoma, debido a que para representar las características de un genotipo se ocuparan muchos alelos.

3.5.2. Codificación real y alfabética

Este tipo de codificación es menos usada que la anterior ya que generalmente los problemas resueltos con AG, son hechos mediante codificación binaria, debido a la facilidad con que se puede manejar el cromosoma. Pero hay ciertos problemas que no se pueden manejar mediante codificación binaria, es por esto que surgen dos tipos de codificación llamadas real y alfabética.

Este tipo de codificación debiera presentar un peor desempeño que su contraparte binaria, debido a que al codificar un cromosoma de n genes reales, se obtienen 2^n posibilidades de cruzamiento, versus las 2^{nm} posibilidades de cruzamiento de un cromosoma que codifique la misma cantidad de números reales de manera binaria, donde m es el número de bits necesarios para representar un número real. Sin embargo, numerosos estudios que comparan ambas codificaciones han mostrado rendimientos superiores de codificaciones reales y alfabéticas en la resolución de problemas tales como la generación de grafos de gramáticas y la evolución de pesos sinápticos de redes neuronales [19].

3.6. Evaluación y Selección

Durante la evaluación, se decodifica el gen, convirtiéndose en una serie de parámetros de un problema, se halla la solución del problema a partir de esos parámetros y se le da una puntuación a esa solución en función de lo cerca que esté de la mejor solución. A esta puntuación se le llama *fitness*. Generalmente los individuos seleccionados para reproducirse son los que poseen un mayor grado de *fitness*, sin embargo, puede que el individuo seleccionado nos lleve a un máximo local y no a un máximo global, es por esto que no deben descartarse los individuos con un menor *fitness*, aprovechando la capacidad exploratoria del algoritmo.

El *fitness* determina siempre los cromosomas que se van a reproducir y aquellos que se van a eliminar, pero hay varias formas de selección de esta función para elegir la población que participara en la creación de la siguiente generación:

- Usar el orden, o rango, y hacer depender la probabilidad de permanencia o evaluación de la posición en el orden.
- Aplicar una operación al *fitness* para escalarlo.
- En algunos casos, el *fitness* no es una sola cantidad, sino diversos números, que tienen diferente consideración. Basta con que tal *fitness* forme un orden parcial, es decir, que se puedan comparar dos individuos y decir cuál de ellos es mejor. Esto suele suceder cuando se necesitan optimizar varios objetivos.

Una vez evaluado el *fitness*, se tiene que crear la nueva población teniendo en cuenta que los buenos rasgos de los mejores se transmitan a esta. Para ello, hay que seleccionar a una serie de individuos encargados de tan ardua tarea. Y esta selección, y la consiguiente reproducción, se pueden realizar de acuerdo a los métodos de selección.

Existen múltiples métodos de selección de los cuales los más usados son ruleta, basado en rangos y por torneo, a continuación se hará referencia a estos métodos [17].

- **Basado en el rango**

En este esquema se mantiene un porcentaje de la población, generalmente la mayoría, para la siguiente generación. Se coloca toda la población por orden de *fitness*, y los M menos dignos son eliminados y sustituidos por la descendencia de alguno de los M mejores con algún otro individuo de la población. A este esquema se le pueden aplicar otros criterios por ejemplo, se crea la descendencia de uno de los paladines/amazonas, y esta sustituye al más parecido entre los perdedores. Esto se denomina *crowding* y fue introducido por De Jong [20]. En realidad, para este esquema se escoge un *crowding factor* (CF). Cuando nace una nueva criatura, se seleccionan CF individuos de la población y se elimina al más parecido a la nueva criatura. Una variante de este es el muestreo estocástico universal, que trata de evitar que los individuos con más *fitness* copen la población en vez de dar la vuelta a una ruleta con una ranura, da la vuelta a la ruleta con N ranuras, tantas como la población; de esta forma, la distribución estadística de descendientes en la nueva población es más parecida a la real.

- **Ruleta**

Se crea un pool genético formado por cromosomas de la generación actual, en una cantidad proporcional a su *fitness*. Si la proporción hace que un individuo domine la población, se le aplica alguna operación de escalado. Dentro de este pool, se cogen parejas aleatorias de cromosomas y se emparejan, sin importar incluso que sean del mismo progenitor (para eso están otros operadores, como la mutación).

$$p(x_j^i) = \frac{f(x_j^i)}{\sum_{k=1}^n f(x_k^i)} \quad (3.1)$$

Este método permite que exista una mayor probabilidad de seleccionar al x_j^i de mayor desempeño, pero dejando abierta la posibilidad de que un x_j^i de bajo rendimiento también sea seleccionado en cada generación, aumentando la diversidad de la población. Sin embargo el método corre el riesgo de perder su capacidad de diversidad y conducir a la convergencia prematura si un grupo pequeño de individuos de x^i poseen una medida de desempeño excesivamente superior a la del resto de individuos.

- **Torneo**

Se escogen aleatoriamente un número x^i de individuos de la población, al obtener los desempeños de los distintos individuos, estos resultados se comparan, así los cromosomas del individuo con mejor puntuación serán usados para la reproducción,

dependiendo del tamaño de la población, existen dos formas de realizar torneo, si la población es pequeña es más probable seleccionar al individuo con valores pequeños respecto de la población, sucede al contrario en poblaciones de tamaño más grande, con esto se busca mejorar la explotación del campo de búsqueda [21].

- **Elitismo**

El elitismo puede verse como una adición a cualquiera de los métodos descritos anteriormente, forzando al método utilizado a retener algunos de los mejores individuos de una población e incluirlos en la siguiente, con el fin de no perder los lugares explorados del espacio de búsqueda que son buenos candidatos. Sin embargo los individuos seleccionados mediante elitismo pueden perderse si no son seleccionados en la siguiente iteración o si se destruyen en el cruzamiento o mutación.

3.7. Aplicaciones de Algoritmos Genéticos

Aunque, como se ha comentado, el Algoritmo que se utilizó en el apartado anterior es muy simple, ha servido para que los estudios realizados en torno a él, se hayan aplicado a diversos problemas y modelos en ingeniería, y en la ciencia en general. Cabe destacar entre ellos:

- **Optimización**

Se trata de un campo especialmente abonado para el uso de los Algoritmos Genéticos, por las características intrínsecas de estos problemas. No en vano fueron la fuente de inspiración para los creadores estos algoritmos. Los Algoritmos Genéticos se han utilizado en numerosas tareas de optimización, incluyendo la optimización numérica y los problemas de optimización combinatoria.

- **Programación automática**

Los Algoritmos Genéticos se han empleado para desarrollar programas para tareas específicas, para diseñar otras estructuras computacionales tales como el autómata celular y las redes de clasificación.

- **Aprendizaje máquina**

Los algoritmos genéticos se han utilizado también en muchas de estas aplicaciones, tales como la predicción del tiempo o la estructura de una proteína. Han servido asimismo para desarrollar determinados aspectos de sistemas particulares de aprendizaje, como pueda ser el de los pesos en una red neuronal, las reglas para sistemas de clasificación de aprendizaje o sistemas de producción simbólica y los sensores para robots.

- **Economía**

En este caso, se ha hecho uso de estos Algoritmos para modelar procesos de innovación, el desarrollo estrategias de puja y la aparición de mercados económicos.

- **Sistemas inmunes**

A la hora de modelar varios aspectos de los sistemas inmunes naturales, incluyendo la mutación somática durante la vida de un individuo y el descubrimiento de familias de genes múltiples en tiempo evolutivo, ha resultado útil el empleo de esta técnica.

- **Ecología**

En la modelización de fenómenos ecológicos tales como las carreras de armamento biológico, la coevolución de parásito-huésped, la simbiosis y el flujo de recursos.

- **Genética de poblaciones**
En el estudio de preguntas del tipo “¿Bajo qué condiciones será viable evolutivamente un gen para la recombinación?”
- **Evolución y aprendizaje**
Los Algoritmos Genéticos se han utilizado en el estudio de las relaciones entre el aprendizaje individual y la evolución de la especie.
- **Sistemas sociales**
En el estudio de aspectos evolutivos de los sistemas sociales, tales como la evolución del comportamiento social en colonias de insectos, la evolución de la cooperación y la comunicación en sistemas multi-agentes.

4. Predictor Neuro-genetico

En capítulos anteriores, se analizaron distintas topologías de redes neuronales y las distintas funciones wavelet como función de activación, definiendo la forma en que estas funcionan y analizando los posibles usos de estas redes. En este capítulo se profundizará el uso de redes neuronales recurrentes de topologías Elman y Jordan, se analizaron también los algoritmos genéticos, cuáles son sus componentes, la forma en que se codifican los cromosomas y de qué manera se eligen los nuevos individuos para generar una población.

En esta sección del documento, se describirá la forma en que pueden interactuar los Algoritmos Genéticos y las redes neuronales de manera que se complementen y nos ayuden a generar una solución óptima para el problema del pronóstico de la anchoveta.

Se utilizaran en primera instancia redes neuronales de Elman y en segunda instancia redes de Jordan, se utilizara Wavelet Mexican Hat o sombrero mexicano como función de activación.

4.1. Algoritmos genéticos y pesos de la red

En general los algoritmos basados en la derivada, son usados para problemas que cuenten con poca variabilidad, una de sus grandes desventajas es quedar atrapado en un óptimo local, debido a esto se utilizaran AG, para la fase de entrenamientos y pruebas de la red.

Al usar redes neuronales existe la posibilidad de que al igual que en la derivada quede atrapado en un máximo local, es por esto que las redes neuronales serán entrenadas por un algoritmo genético, esto quiere decir que las conexiones sinápticas en las neuronas serán variables, para que esto pueda ser factible la red neuronal necesitara estar codificada de manera que sea usable por el AG, esta codificación se hace mediante cromosomas. El cromosoma está formado por un conjunto de los alelos, que a su vez pueden tomar los genes es un subconjunto de los números reales, así se crea una población de cromosomas aleatoria que evoluciona en el tiempo, para ver el desempeño del algoritmo midiendo el desempeño del algoritmo.

Este método generalizado puede modificarse para dar solución a un caso particular mediante el uso de wavelets, definiendo sus parámetros de escala (*dilatación y contracción de la señal*) y la función de traslación, se utilizaran dos tipos de activación la de las neuronas de capa oculta y la de wavelet, debido a las funciones a usar y a la recurrencia que se tiene.

En el caso de la red de Elman tiene tantas neuronas de contexto como ocultas, por lo que se contara con dos cromosomas uno que pertenecerá a las neuronas de contexto y a su vez uno que marque las entradas con los datos correspondientes, así en una primera aproximación si se

tienen cinco neuronas de contexto, es porque en la capa oculta se tienen cinco neuronas así primer cromosoma de contexto seria:

0	0	0	0	0
-----	-----	-----	-----	-----

FIGURA 4-1 : CROMOSOMA DE CONTEXTO

Donde cada alelo del cromosoma de contexto estará definido por la Formula 2.6 y se irá modificando a medida que se vaya iterando, este cromosoma no crecerá pero sus valores se modificaran.

Se debe contar además con un cromosoma en el que se estarán las entradas, como el que se mostrara a continuación.

X_i	X_{i+1}	X_{i+2}	X_{i+1}	$\dots X_{i+n}$
-------	-----------	-----------	-----------	-----------------

FIGURA 4-2 : CROMOSOMA DE PESOS

Es así como se tendrán todas las entradas en un mismo cromosoma, y la venta temporal es la que ira variando. Una de las posibilidades es trabajar estos cromosomas por separado o juntarlos en unos solo como lo indica la Formula 2.7 siempre que las neuronas de capa oculta sean fijas y que la cantidad de entradas no varié. Con este nuevo cromosoma se puede calcular la función de activación de las demás neuronas.

X_i	X_{i+1}	X_{i+2}	X_{i+1}	$\dots X_{i+n}$	C_i	C_{i+1}	C_{i+2}	C_{i+3}	$\dots C_{i+n}$
-------	-----------	-----------	-----------	-----------------	-------	-----------	-----------	-----------	-----------------

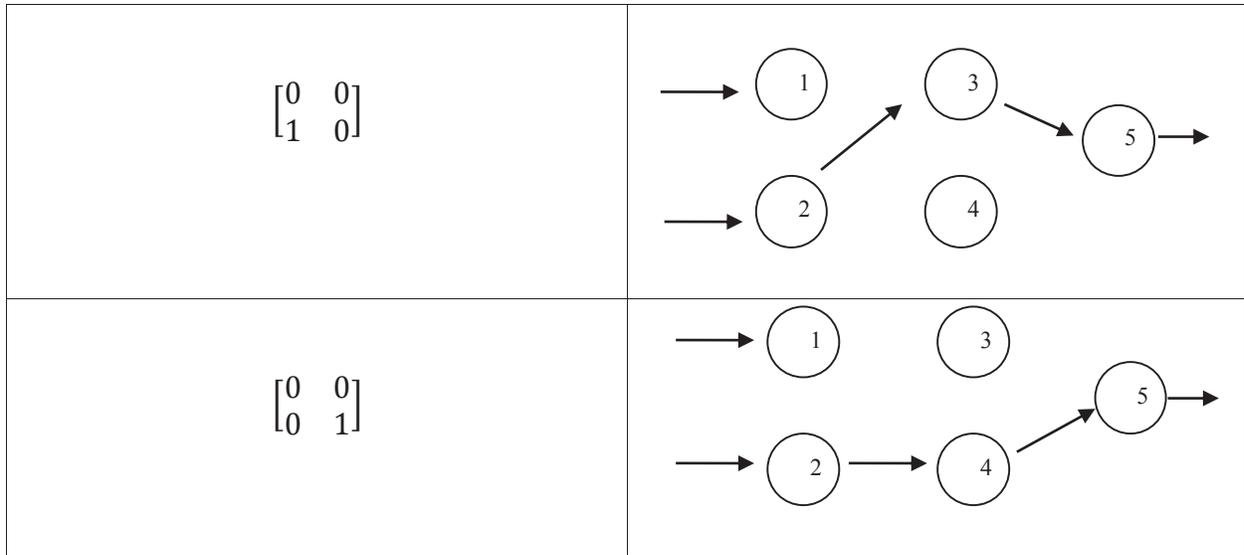
Figura 4-3 : Cromosoma Final

4.2. Determinando la Topología de la red

En nuestro caso los AG serán usados para la determinación del número de neuronas artificiales a utilizar, además de las interconexiones que se observan en la red, dado que generalmente éstos se deciden a priori (*Random*) y se modifican mediante ensayo-error.

TABLA 4-1 : CODIFICACIÓN DIRECTA DE LA TOPOLOGÍA DE UNA RED NEURONAL

Codificación	Topología
$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	
$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$	



La forma de codificación puede ser directa, esto quiere decir que se hace mediante una matriz binaria cuadrada donde cada elemento de esta matriz puede tomar valores de verdadero o falso, para el caso de verdadero, que es donde existe una conexión entre neuronas se usara el valor 1, al contrario si no existe relación alguna entre estas, el valor a utilizar será el 0. Diferentes pruebas han demostrado que esta forma de codificación encuentra con relativa facilidad arquitecturas que luego de ser entrenadas, se obtienen resultados con una menor tasa de error.

4.3. Modelo de funcionamiento de la Red

De acuerdo a los datos proporcionados por Serna pesca se pudo hacer el grafico de la figura 1-1, en esta se aprecia que la información es variada presentando muchos picos y valles, esto quiere decir frecuencias muy altas y también bajas de la data, es por esto que es necesario realizar un suavizado sobre ella, mediante algún filtro wavelet, de manera de poder discretizar la data y obtener una mejor aproximación a lo que será la entrada de la red. En una primera instancia se ocuparan estos datos sin suavizar para realizar una primera aproximación a la solución, es por esto que se ha propuesto un modelo, mostrado a continuación.

Básicamente la red neuronal funciona de la siguiente manera:

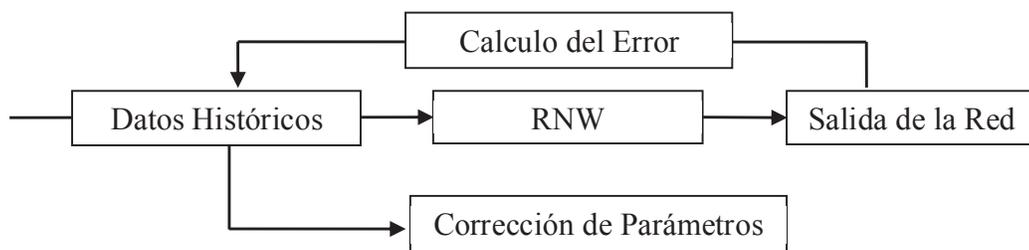


FIGURA 4-4 : DIAGRAMA DE BLOQUES RNA

El segundo modelo propuesto es similar al primero, la diferencia entre ambos es que en este último se aplica el filtro para bajas wavelet, separando la señal en alta y baja, dejando la alta de lado por interferir con el correcto funcionamiento del algoritmo.

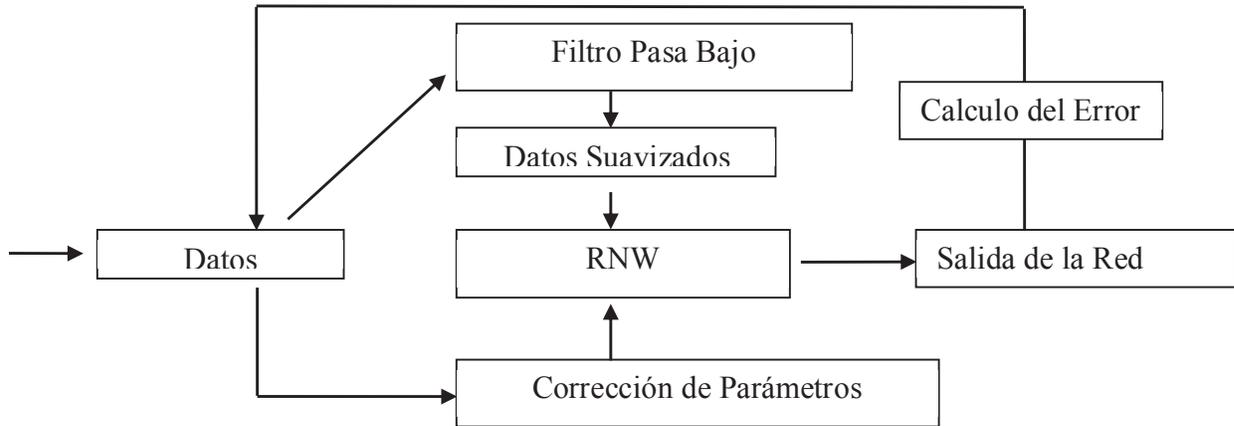


FIGURA 4-5 : DIAGRAMA DE BLOQUES RNA CON FILTRO DE SUAVIZADO

4.4. Desempeño de una RNA

Al momento de evaluar el desempeño que ha tenido la red neuronal, surgen múltiples soluciones tales como la función *fitness*, o el cálculo del coeficiente de determinación, aun siendo estos buenos métodos de medida del desempeño, la más conocida de estas el cálculo del error cuadrático medio, este nos permite cuantificar el error de lo estimado en la misma unidad de medida que variables de predicción. [22]

$$RMSE = \sqrt{\frac{\sum_{t=1}^N (Q_t - \widehat{Q}_t)^2}{N}} \quad (4.1)$$

Utilizamos también el MAPE o error absoluto porcentual promedio, la cual es una medida de precisión de un método para evaluar valores de series de tiempo en las estadísticas, específicamente en la estimación de tendencia. Por lo general, la precisión expresa como un porcentaje, y se define por la fórmula:

$$MAPE = \frac{100}{n} \sum_{i=0}^n \left| \frac{\hat{y}_i - y_i}{\hat{y}_i} \right| \quad (4.2)$$

Donde \hat{y}_i es el valor real y el valor de y_i es el pronóstico.

La diferencia entre \hat{y}_i y y_i se divide por el valor real de nuevo. El valor absoluto en este cálculo se suma para cada punto equipada o previsto en tiempo y se divide de nuevo por el número de puntos ajustados n. Multiplicando por 100 lo convierte en un porcentaje de error.

Aunque el concepto de MAPE suena muy simple y convincente, tiene un gran inconveniente en su aplicación, ya que si existen valores de ceros esperados habrá una división por cero. Luego si tenemos un ajuste perfecto, MAPE es cero.

Con el fin de calcular la ganancia que tiene el mejor modelo encontrado en las pruebas realizadas se propone la siguiente ecuación 4.8.

$$Ganancia = \left(\frac{M1 - M2}{M1} \right) * 100 \quad (4.3)$$

Donde M1 corresponde al modelo 1 con mejor resultado respecto al modelo 2 que se desea comparar.

4.5. Elección de padres

La selección de los padres se hace a través de la función de selección más utilizada, es aquella que es proporcional a la función objetivo. Su funcionamiento consiste en que cada individuo de una generación tiene una probabilidad de ser seleccionado proporcional al valor de su función objetivo.

$$p_{j,t}^{prop} = \frac{g(I_t^j)}{\sum_{j=1}^{\lambda} g(I_t^j)} \quad (4.4)$$

Esta representa la probabilidad que el sujeto I_t^j sea seleccionado como padre.

Existen además otros métodos de selección, que fueron explicados en la sección 3.6 de este texto, pero de aquí en adelante usaremos la elección de *padres por rango* para ambas topologías de red.

4.6. Generar nueva población

Una vez que ya se tienen los individuos se procederá al proceso de interacción, proceso descrito en pseudo lenguaje con anterioridad, para esto, se procede con el proceso de cruzamiento y mutado hasta elegir los padres que generara nuevos hijos.

Es por esto que el cruzamiento estará basado en uno o más puntos de mutación de manera de brindar una mayor diversidad a la población, estos serán seleccionados aleatoriamente de entre los alelos del cromosoma, incluyendo el origen, con una probabilidad de elección de un **10%**. Es muy importante la elección de los padres ya que ellos generaran la próxima generación de hijos, esta pareja generara una pareja a su vez de cromosomas, estos nuevos cromosomas serán llamados hijos. Del padre seleccionado aleatoriamente se debe mutar su alelo reemplazándolo por un alelo real generado aleatoriamente en el intervalo **[-1 1]**,

una vez finalizado este proceso, los hijos o nuevos individuos son agregados a la población y se repite este proceso hasta ocupar todos los datos proporcionados.

Con los resultados obtenidos al procesar estos datos se podrán crear tantos hijos como sea necesario, para poder optar a un espacio de búsqueda mayor y poder encontrar una solución lo más cercano a lo óptimo. El proceso de generación de población se realizara hasta que los resultados sean cercanos a los esperados. El resultado esperado depende no solo del proceso de mutación y de generación de nueva población, importante también es la cantidad de neuronas que están presentes dentro del modelo, las capas existentes y de la calidad de la data.

5. Discusión de Resultados

5.1. Introducción

En esta sección se analizarán los resultados del desarrollo de los algoritmos propuestos en el capítulo anterior tanto de Elman como de Jordán. En primer lugar se debe recordar que la función de desempeño de la red wavelet recurrente del tipo Elman y Jordán será el coeficiente de determinación r^2 , a su vez r está definida en la Formula 4.1, para esto se tiene que utilizar la minimización de la sumatoria de X_i .

$$\sum_{i=1}^n x_i^2 \quad (5.1)$$

Al implementar una RNA wavelet recurrente se tiene que al entrenarla y luego inyectarla con datos, podemos utilizar para medir en la data de salida, la minimización del error cuadrático medio, que es la función utilizada como medio, para comprobar el grado de adaptación del AG a la red. Los tamaños de las poblaciones variaron de acuerdo a un valor arbitrario de individuos, para después utilizar el doble del largo del cromosoma como largo de la población.

A continuación se muestra la tabla con los resultados obtenidos en la fase de validación de las redes, indicando el error cuadrático medio (*RMSE*) obtenido en cada prueba y su coeficiente de determinación (R^2), utilizado para describir la proporción de la varianza total en los datos observados que pueden explicarse mediante el modelo.

Se realizaron pruebas con diferentes tamaños de población y con un número diferente de generaciones cada vez, con el objetivo de lograr obtener un valor óptimo, una vez obtenido este valor se estandariza para poder realizar la comparativa, estos resultados serán detallados a continuación.

Para las pruebas del algoritmo de Jordán, al igual que el de Elman, se utilizó una arquitectura que consta de ocho nodos en la capa de entrada, diez nodos en la capa oculta y un nodo en la capa de salida (*8in-10hi-1out*), se llegó a este valor luego de realizar diversas pruebas y obtener mejores resultados con esta configuración. Dentro de las pruebas realizadas, se hicieron de forma iterativa e incremental, comenzando con cien individuos y cien generaciones. En una próxima fase de prueba se aumentaron los individuos en cien, dejando las generaciones estáticas, para así poder comparar el resultado obtenido al incrementar la cantidad de individuos dentro de la misma configuración. Una vez realizado el análisis correspondiente, se procede a general los gráficos correspondientes, en donde se comparan los resultados obtenidos, con los resultados reales, además de realizar un gráfico de dispersión,

este nos ayuda a visualizar que tan alejados están los resultados de la recta esperada. Así como se mantuvieron las generaciones estáticas, esta vez serán las que varíen y se dejen los individuos en cien. Se realizarán por tanto las pruebas correspondientes y una vez estén listos se procede a realizar el análisis correspondiente.

Tal como se indicó en un principio, este proceso será iterativo con lo que se obtendrán diversas configuraciones que varían en generaciones y en población.

5.2. Procedimiento

A continuación se muestran las gráficas de convergencia del algoritmo neuro-genético, usando como configuración inicial para todas las pruebas ocho nodos en la capa de entrada, diez nodos en la capa oculta y el nodo correspondiente en la capa de salida (*8in-10hi-1out*); Esta configuración fue la resultante óptima luego del entrenamiento de la red; Para esto la red fue sometida a muchas fases de prueba, resultando la ventana de mejores resultados, con configuraciones de población partiendo de las cien, doscientos, trescientos llegando a cuatrocientos, una vez realizadas las iteraciones se procede a realizar un gráfico de convergencia, este nos ayuda a observar de manera gráfica cual configuración para el tamaño de población dada se acerca más rápidamente a la solución, utilizando este y además otros indicadores se podrá determinar cuál es la mejor configuración del algoritmo de forma que la convergencia sea más rápida hacia la solución deseada, en nuestro caso y por el motivo de este trabajo, se busca obtener como solución el tonelaje de captura de anchovetas para su extracción en un periodo determinado. En los resultados obtenidos, la mejor configuración se obtuvo con doscientos generaciones, por lo que la ventana de resultados tendrá una variación en la población pero con generaciones estáticas.

Una vez los resultados de estas nuevas configuraciones están listos, se procede a realizar un análisis comparativo a través de tablas en donde se verá cuales fueron resultados obtenidos. De esta forma y a través de los indicadores se podrá determinar cuál fue el rendimiento de cada configuración del algoritmo, cual converge de mejor manera y con la elección de este se procura a realizar una nueva iteración.

Una vez seleccionada la configuración con la que el algoritmo converge más rápidamente, se realizan nuevas pruebas, pero esta vez lo que varía no es la población, ni las generaciones, sino que la topología del algoritmo. La configuración inicial que se dará a la población seleccionada será 8IN-10HID, en las próximas iteraciones variarán las HID la que será incrementada hasta encontrar la configuración adecuada para tener una mejor convergencia de estos algoritmos.

5.3. Análisis de convergencia y configuración

El análisis de convergencia se basa en el RMSE, este nos indica que tan alejados están los valores obtenidos por el modelo de los datos reales. El gráfico de la figura 5-1, corresponde al análisis de convergencia realizado a las configuraciones antes señaladas, con doscientos generaciones cada una de las pruebas, para la topología de Jordán y con una

población inicial de cien individuos creciendo hasta llegar a los cuatrocientos así en la figura 5-2 se verá el grafico de convergencia para el modelo de Elman con la misma cantidad de generaciones y la misma población inicial - final.

Así y como se explicó en el procedimiento, se seleccionó el mejor modelo tanto para Jordán como para Elman, esta elección se basó en la convergencia, el que converja de mejor a la solución deseada será el elegido. De las 4 pruebas realizadas y mostradas a continuación, el modelo seleccionado es el de doscientos individuos y doscientos generaciones denotadas por 200p-200 g.

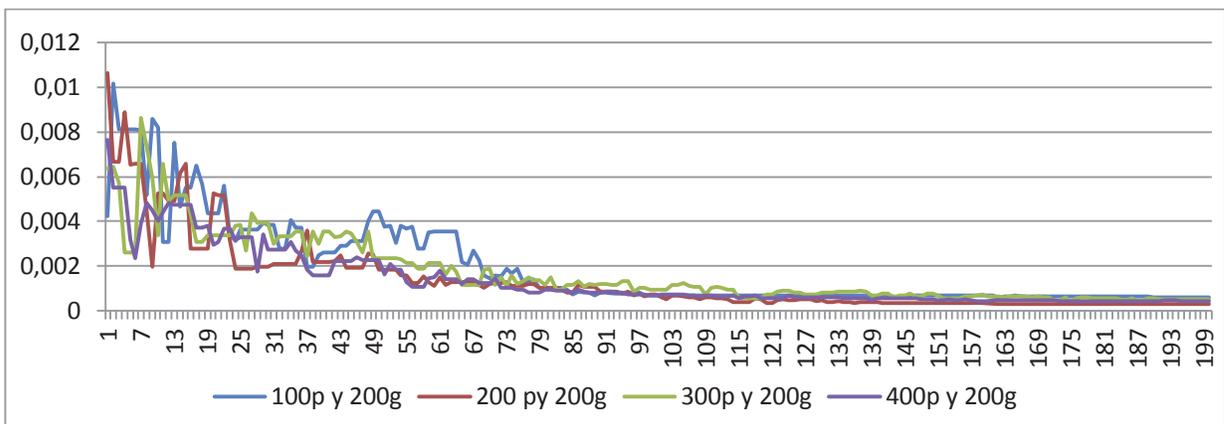


FIGURA 5-1 : GRAFICO DE CONVERGENCIA MODELO DE JORDAN

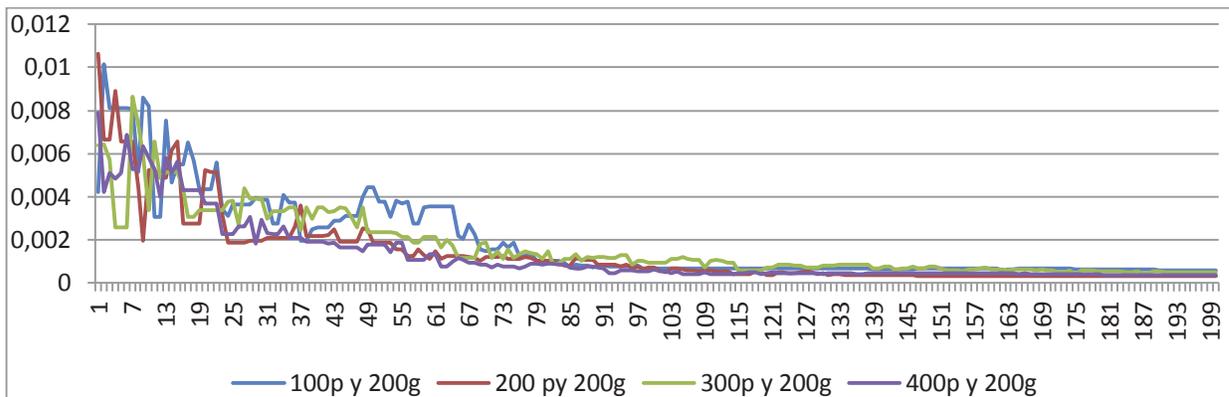


FIGURA 5-2 : GRAFICO DE CONVERGENCIA MODELO DE ELMAN

Realizando una comparativa de convergencia entre los modelos de Elman y Jordán se tiene que la convergencia es mejor en el modelo de Jordán que en el modelo de Elman, se espera corroborar esto en análisis posteriores.

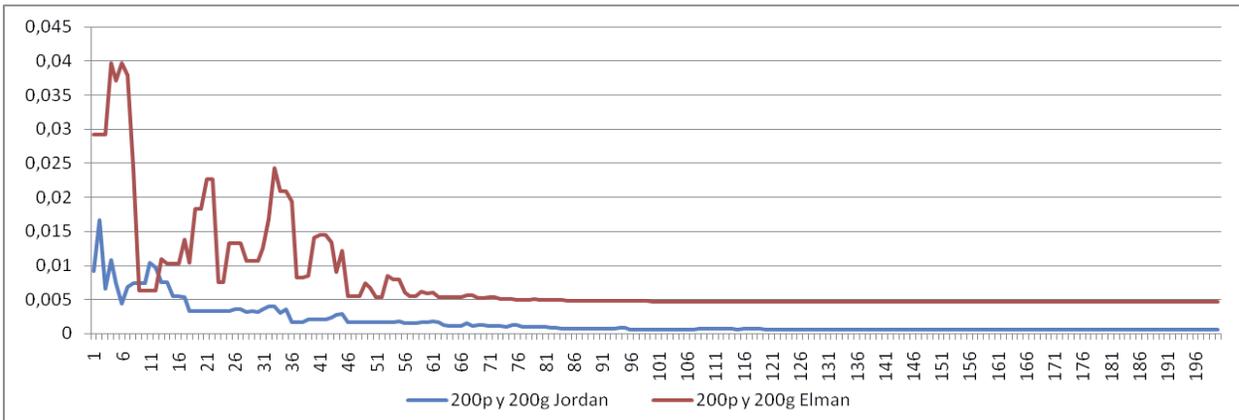


FIGURA 5-3 : GRAFICO DE CONVERGENCIA ENTRE MODELOS ELMAN Y JORDAN

De los resultados obtenidos en cada etapa de entrenamiento se obtuvieron gráficos de dispersión y de comparación entre datos estimados y observados.

5.3.1. Graficas Obtenidas Modelos Elman y Jordan

A continuación, se mostraran las tablas obtenidas de la fase de validación de ambos modelos, que se determinan a partir de la selección de la convergencia del mejor modelo. Para las gráficas a continuación la **línea azul** representara la curva de datos estimada, en cambio la **línea roja**, representara la curva de datos observada.

Graficas del modelo de Jordan

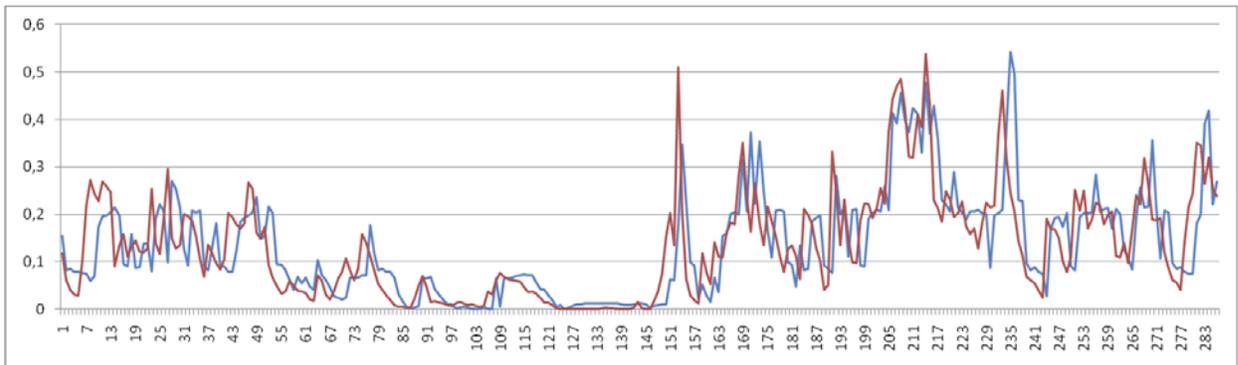


FIGURA 5-4 : GRAFICO DE CONVERGENCIA MODELO DE JORDAN 100P-200G

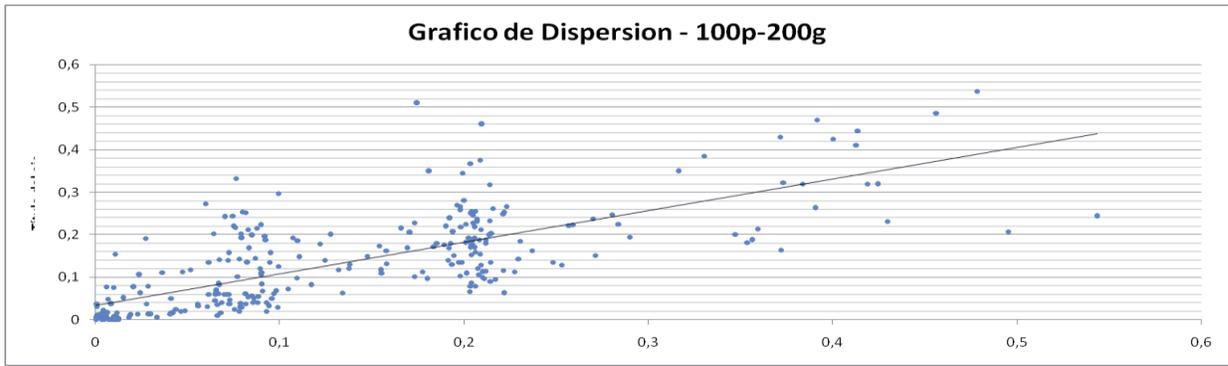


FIGURA 5-5 : GRAFICO DE DISPERSIÓN MODELO DE JORDAN 100P-200G

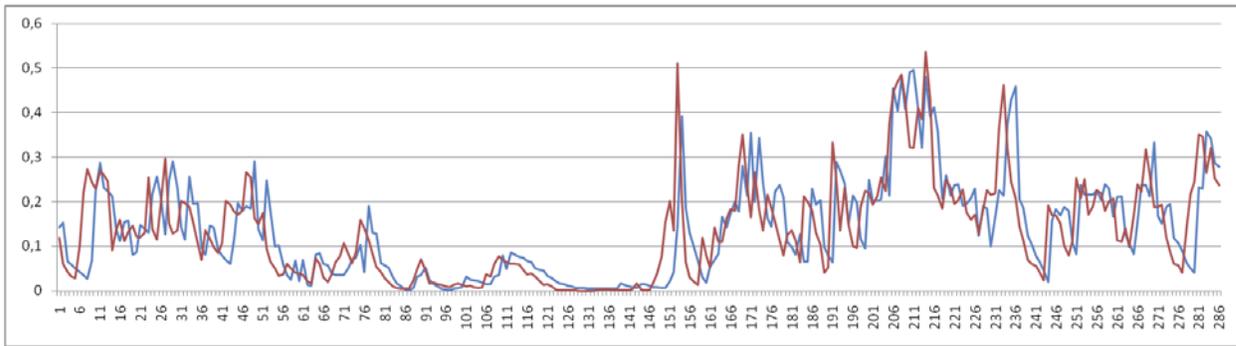


FIGURA 5-6 : GRAFICO DE CONVERGENCIA MODELO DE JORDAN 200P-200G

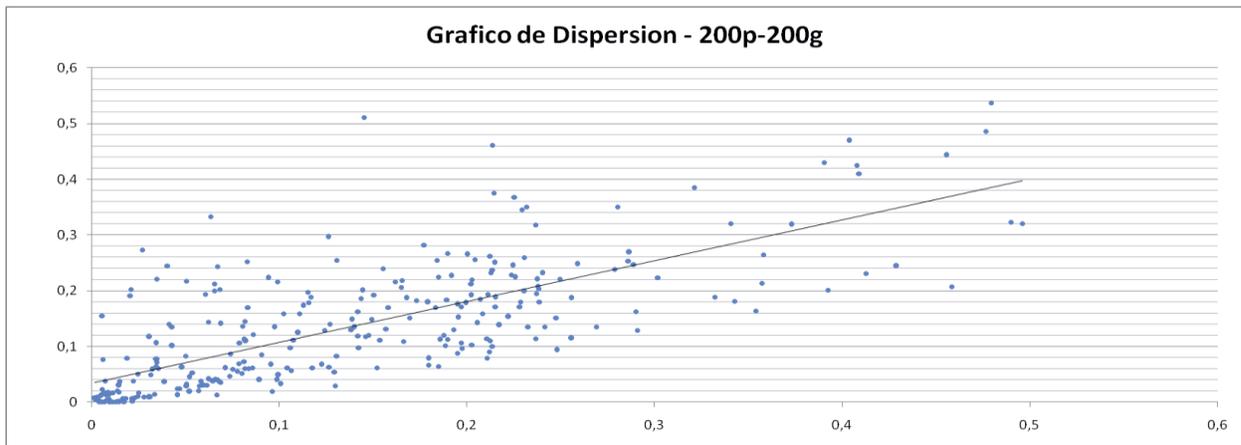


FIGURA 5-7 : GRAFICO DE DISPERSIÓN MODELO DE JORDAN 200P-200G

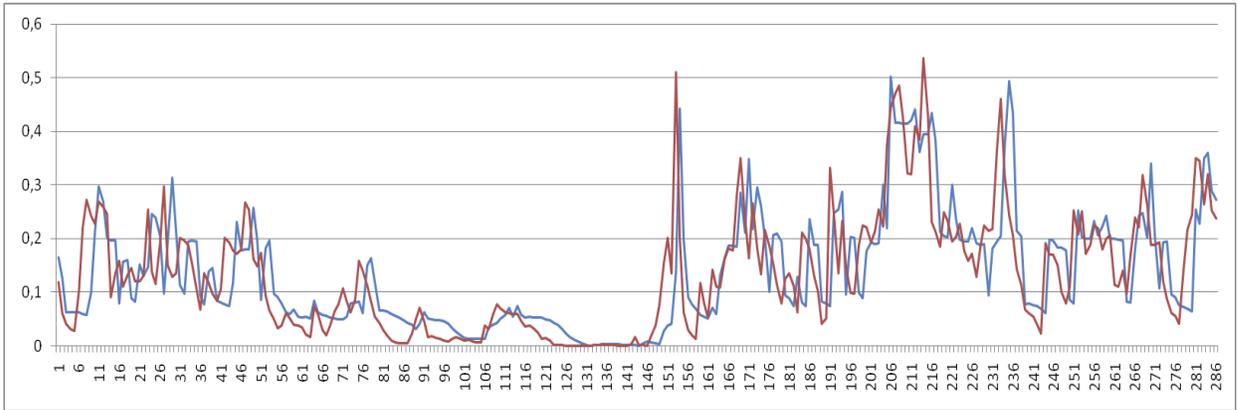


FIGURA 5-8 : GRAFICO DE CONVERGENCIA MODELO DE JORDAN 300P-200G

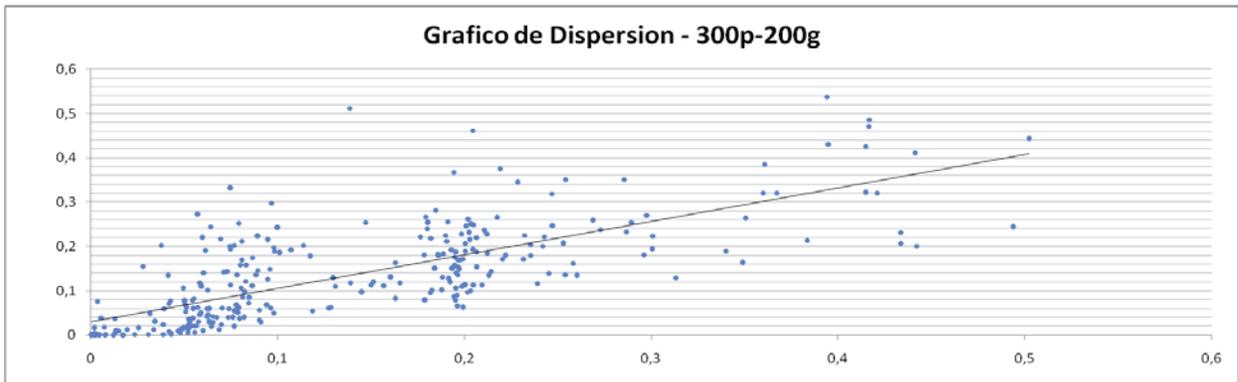


FIGURA 5-9 : GRAFICO DE DISPERSIÓN MODELO DE JORDAN 300P-200G

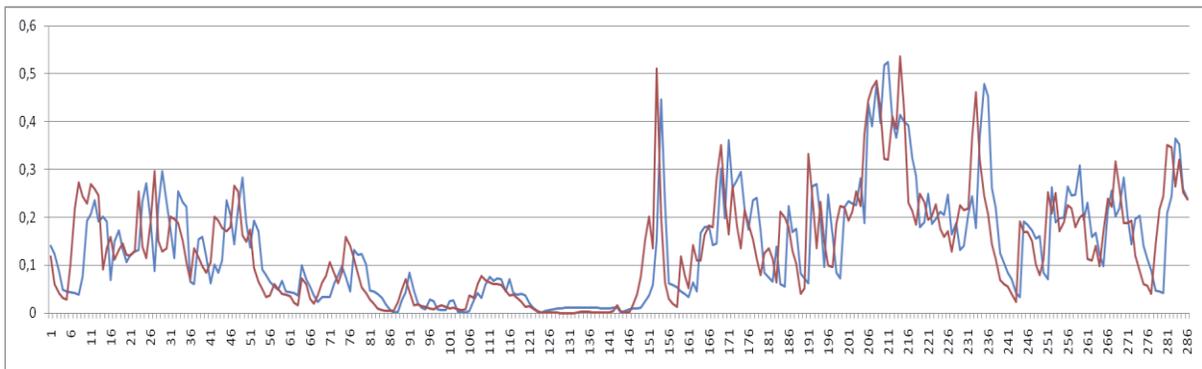


FIGURA 5-10 : GRAFICO DE CONVERGENCIA MODELO DE JORDAN 400P-200G

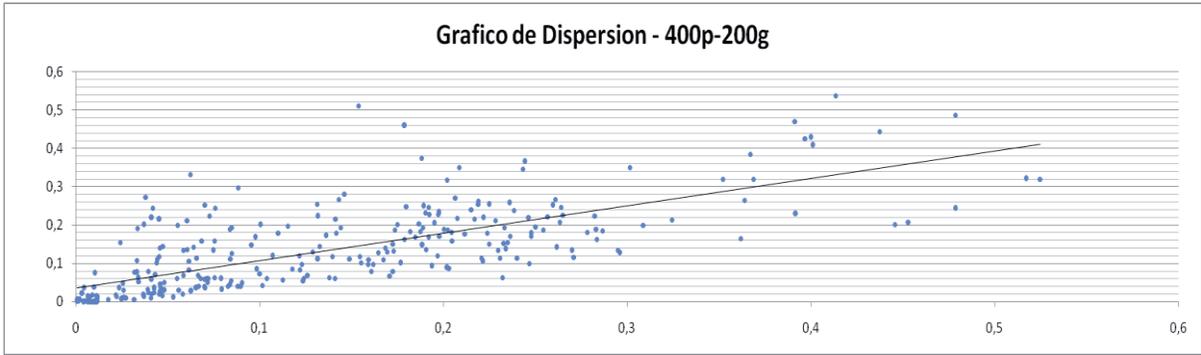


FIGURA 5-11 : GRAFICO DE DISPERSIÓN MODELO DE JORDAN 400P-200G

Graficas del modelo de Elman

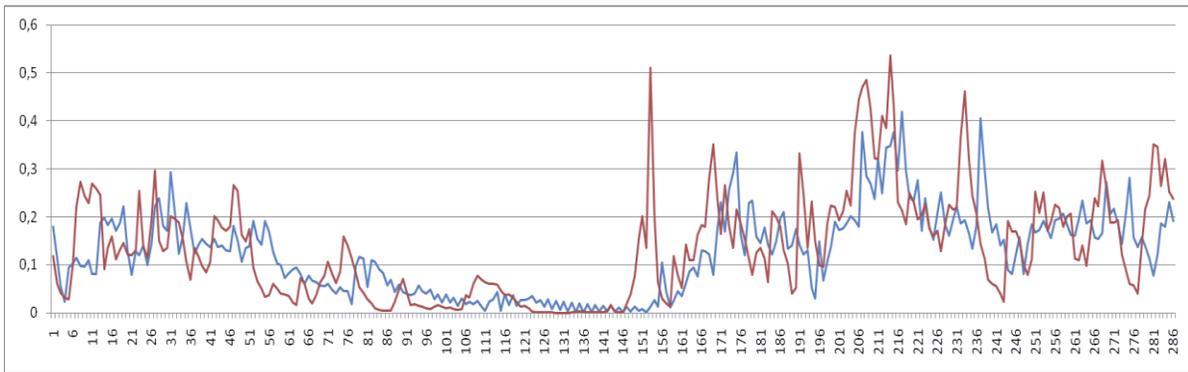


FIGURA 5-12 : GRAFICO DE CONVERGENCIA MODELO DE ELMAN 100P-200G

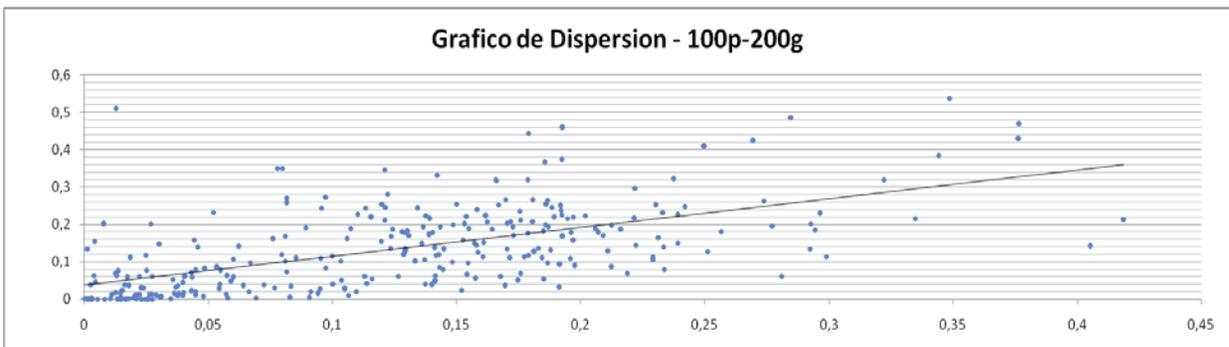


FIGURA 5-13 : GRAFICO DE DISPERSIÓN MODELO DE ELMAN 100P-200G

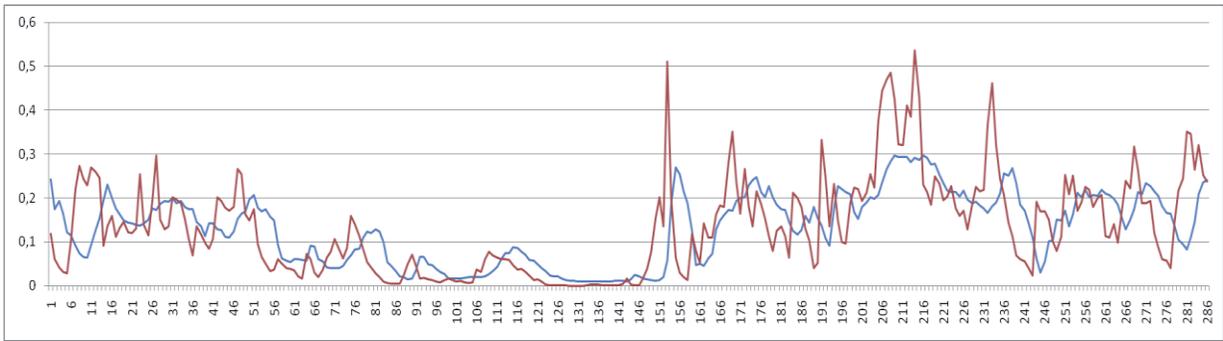


FIGURA 5-14 : GRAFICO DE CONVERGENCIA MODELO DE ELMAN 200P-200G

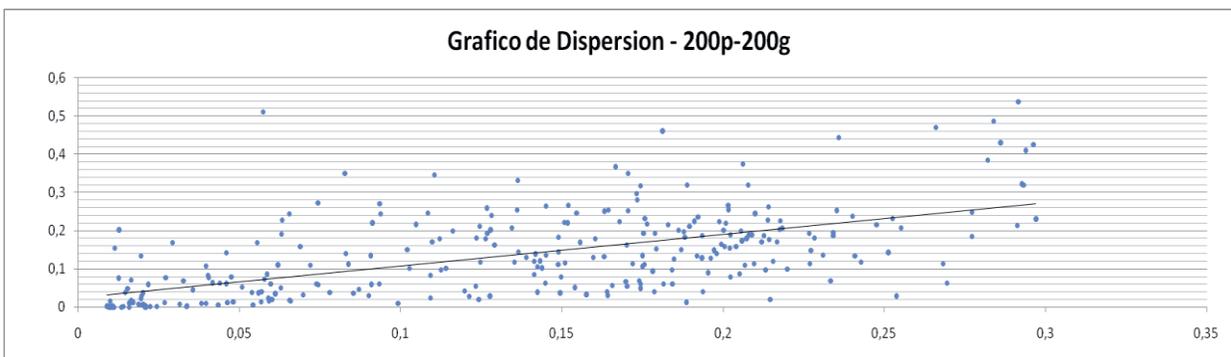


FIGURA 5-15 : GRAFICO DE DISPERSIÓN MODELO DE ELMAN 200P-200G

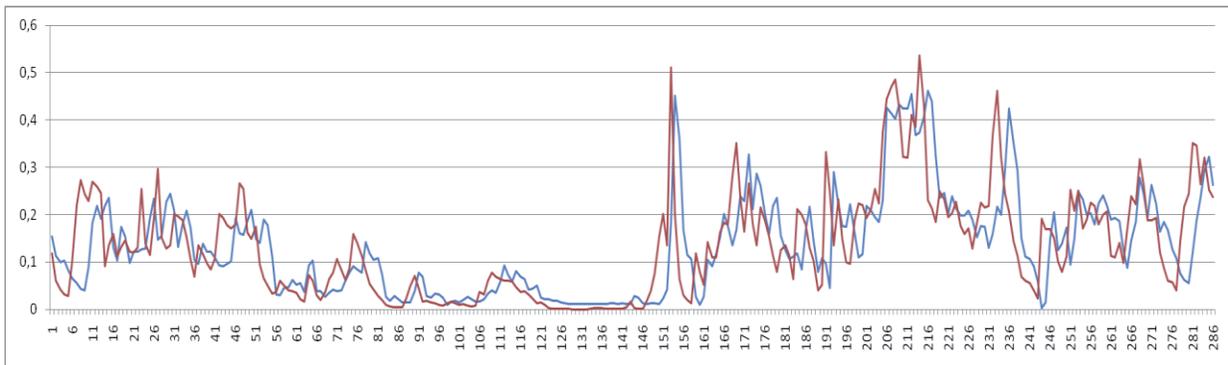


FIGURA 5-16 : GRAFICO DE CONVERGENCIA MODELO DE ELMAN 300P-200G

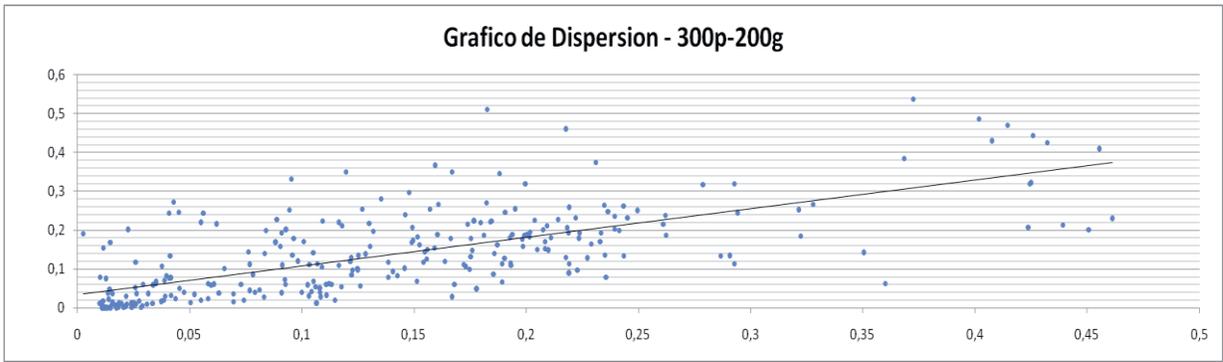


FIGURA 5-17 : GRAFICO DE DISPERSIÓN MODELO DE ELMAN 300P-200G

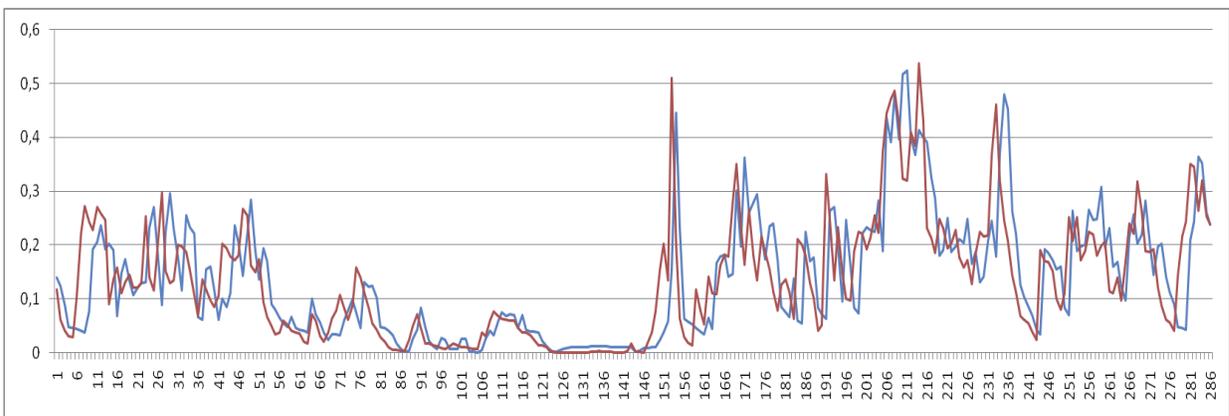


FIGURA 5-18 : GRAFICO DE CONVERGENCIA MODELO DE ELMAN 400P-200G

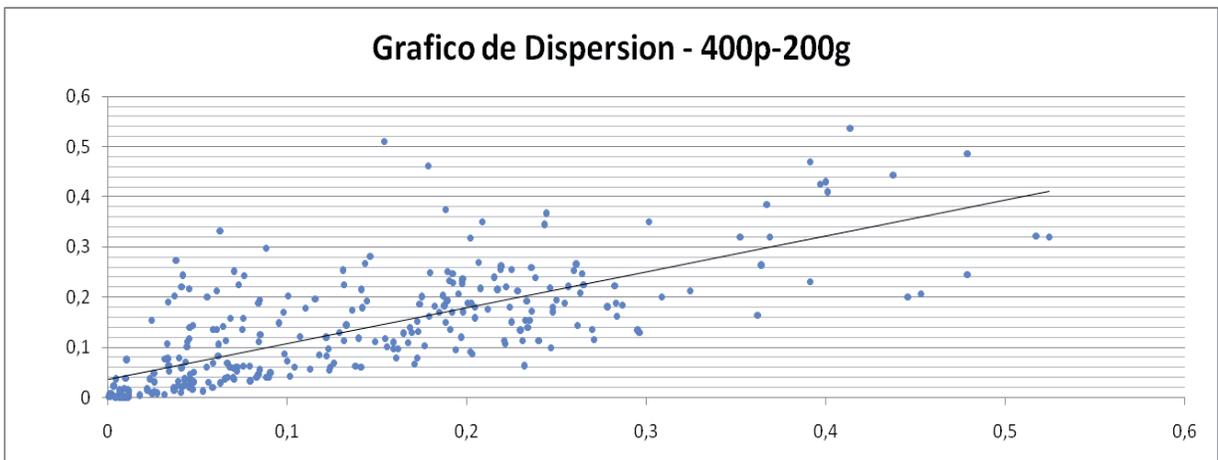


FIGURA 5-19 : GRAFICO DE DISPERSIÓN MODELO DE ELMAN 400P-200G

5.3.2. Modelo de Jordan

En esta sección se procede a mostrar los resultados obtenidos dentro de las cuatro fases etapas de prueba realizadas al modelo. Para una primera iteración de las pruebas se utilizó de configuración ocho neuronas de entrada denominada en la tabla por **8IN**, diez neuronas en capa oculta, denotada por **10HID**, esta última varia, entre diez y trece, es por esto que se muestran cuatro tablas a continuación, mostrándose además los indicadores correspondientes al RMSE, MAPE, R^2 , la cantidad de parámetros de la red y el tiempo que demora en realizarse cada iteración.

TABLA 5-1: RESULTADOS OBTENIDOS CONFIGURACIÓN 8IN-10HID PARA JORDAN

8IN-10HID	RMSE	MAPE	R^2	PAR	SEG.
1	0,033683972	1,20591811	0,91040245	121	1294
2	0,039033191	1,53484613	0,88273871	121	1105
3	0,02295452	1,58246886	0,9590056	121	1263
4	0,035807681	2,65279808	0,90066312	121	1105
5	0,07370346	1,37895102	0,60929863	121	1187
Promedios	0,044496067	1,67099644	0,8524217	121	1190,8
Máximo	0,07370346	2,65279808	0,9590056	121	1294
Mínimo	0,02295452	1,20591811	0,60929863	121	1105
Des. Estándar	0,044324147	0,56830821	0,13881379	0	87,4654217

TABLA 5-2 : RESULTADOS OBTENIDOS CONFIGURACIÓN 8IN-11HID PARA JORDAN

8IN-11HID	RMSE	MAPE	R^2	PAR	SEG.
1	0,028783676	1,0923021	0,9347224	133	1107
2	0,02678619	1,0219919	0,9447132	133	1023
3	0,032813107	2,6481084	0,9187328	133	1087
4	0,026758176	0,5031134	0,9434549	133	1030
5	0,028891175	1,31637899	0,93540585	133	1062
Promedios	0,028891175	1,31637899	0,93540585	133	1061,8
Máximo	0,032813107	2,6481084	0,9447132	133	1107
Mínimo	0,026758176	0,50311349	0,91873281	133	1023
Des. Estándar	0,012124356	0,80183823	0,01036667	0	36,03748049

TABLA 5-3 : RESULTADOS OBTENIDOS CONFIGURACIÓN 8IN-12HID PARA JORDAN

8IN-12HID	RMSE	MAPE	R^2	PAR	SEG.
1	0,032927192	0,8403806	0,9146081	145	1206
2	0,036120631	1,0266275	0,9035742	145	1293
3	0,030967725	1,3325748	0,9265000	145	1211
4	0,02356056	1,4105055	0,9575085	145	1252
5	0,030162891	1,8843022	0,9288719	145	1261
Promedios	0,031025796	1,2988781	0,9262126	145	1244,6
Máximo	0,036120631	1,8843022	0,9575085	145	1293
Mínimo	0,02356056	0,8403806	0,9035742	145	1206
Des. Estándar	0,016555966	0,4002055	0,0202098	0	36,3496905

TABLA 5-4 : RESULTADOS OBTENIDOS CONFIGURACIÓN 8IN-13HID PARA JORDAN

8IN-13HID	RMSE	MAPE	R^2	PAR	SEG.
1	0,024450153	0,66134679	0,95277934	157	1206
2	0,021960874	1,2268062	0,9628655	157	1293
3	0,033485519	1,30178919	0,92239963	157	1235
4	0,030223666	2,46457826	0,92964735	157	1270
5	0,024928698	1,31980821	0,95133655	157	1250
Promedios	0,027335874	1,39486573	0,94380567	157	1250,8
Máximo	0,033485519	2,46457826	0,9628655	157	1293
Mínimo	0,021960874	0,66134679	0,92239963	157	1206
Des. Estándar	0,016205863	0,65667264	0,01702348	0	33,19186647

De las pruebas realizadas obtenemos un promedio y con este tener una idea más acabada del modelo de Jordán con respecto a sus indicadores. Al obtener el promedio de estos se tiende a estandarizar el modelo.

TABLA 5-5 : PROMEDIO DE LAS PRUEBAS REALIZADAS AL MODELO JORDAN 200P-200G

Configuración	RMSE	MAPE	R^2	PAR	SEG.
Promedios	0,033632127	1,42027983	0,91446146	139	1188,85
Max	0,047262564	2,41244675	0,95602322	139	1246,75
Min	0,023876348	0,80268977	0,83850134	139	1135

5.3.3. Modelo de Elman

Al igual que en modelo de Jordan, se realizaron cuatro pruebas del modelo con la misma cantidad de neuronas de entrada y de neuronas en la capa oculta, la notación es igual a la anterior. A continuación se muestran los resultados obtenidos en estas pruebas realizando cinco iteraciones en cada una. La cantidad de parámetros es mayor que en el Jordan debido a que en Elman se cuenta con reinyección en la capa oculta, esto quiere decir que deben existir

igual cantidad de neuronas de contexto que las que existen en la capa oculta, cosa que no ocurría con el modelo anterior.

TABLA 5-6 : RESULTADOS OBTENIDOS CONFIGURACIÓN 8IN-10HID PARA ELMAN

8IN-10HID	RMSE	MAPE	R ²	PAR	SEG.
1	0,075286652	3,87962455	0,56304259	190	484
2	0,074700535	3,61444523	0,56226022	190	423
3	0,060261762	1,74024257	0,71860378	190	456
4	0,080439978	7,85156993	0,51051427	190	441
5	0,085979474	1,17041312	0,49149426	190	439
Promedios	0,075819259	3,65125908	0,56918302	190	448,6
Máximo	0,085979474	7,85156993	0,71860378	190	484
Mínimo	0,060261762	1,17041312	0,49149426	190	423
Des. Estándar	0,037297453	2,62215074	0,08928852	0	22,98477757

TABLA 5-7 : RESULTADOS OBTENIDOS CONFIGURACIÓN 8IN-11HID PARA ELMAN

8IN-11HID	MSE	MAPE	R ²	PAR	SEG.
1	0,107059703	12,2536205	0,3604576	220	915
2	0,056061662	1,52731007	0,75348017	220	923
3	0,078761221	3,7606642	0,54030737	220	1017
4	0,091437684	7,79429788	0,40862705	220	906
5	0,059149134	2,50345602	0,58269346	220	983
Promedios	0,080830069	5,56786973	0,52911313	220	948,8
Máximo	0,107059703	12,2536205	0,75348017	220	1017
Mínimo	0,056061662	1,52731007	0,3604576	220	906
Des. Estándar	0,058991864	4,43393141	0,15515985	0	48,63332191

TABLA 5-8 : RESULTADOS OBTENIDOS CONFIGURACIÓN 8IN-12HID PARA ELMAN

8IN-12HID	RMSE	MAPE	R ²	PAR	SEG.
1	0,067039615	1,86467975	0,65035087	252	1013
2	0,068989709	1,86598819	0,62604601	252	1049
3	0,081722274	3,12462845	0,56146113	252	983
4	0,069877965	0,91752918	0,63598394	252	917
5	0,075515892	1,75238187	0,6097736	252	927
Promedios	0,07282582	1,90504149	0,61672311	252	977,8
Máximo	0,081722274	3,12462845	0,65035087	252	1049
Mínimo	0,067039615	0,91752918	0,56146113	252	917
Des. Estándar	0,029854313	0,78883333	0,03424509	0	56,15336143

TABLA 5-9 : RESULTADOS OBTENIDOS CONFIGURACIÓN 8IN-13HID PARA ELMAN

8IN-13HID	RMSE	MAPE	R^2	PAR	SEG.
1	0,060784538	1,22471206	0,70836436	286	1638
2	0,060630108	0,73531541	0,71879031	286	1399
3	0,060627551	2,53004819	0,71984232	286	1647
4	0,054686653	1,62262524	0,77078424	286	1448
5	0,07966781	2,80443078	0,53778918	286	1453
Promedios	0,063849902	1,78342634	0,69111408	286	1517
Máximo	0,07966781	2,80443078	0,77078424	286	1647
Mínimo	0,054686653	0,73531541	0,53778918	286	1399
Des. Estándar	0,036109832	0,87127054	0,08908524	0	116,5354023

TABLA 5-10 : PROMEDIO DE LAS PRUEBAS REALIZADAS AL MODELO ELMAN 200P-200G

Configuración	RMSE	MAPE	R^2	PAR	SEG.
Promedios	0,073590896	3,22689916	0,60153334	237	975,85
Max	0,089274464	6,50856241	0,72330477	237	1039,5
Min	0,059706197	1,08764194	0,48780054	237	914,5

5.4. Comparativa de Modelos Predictivos

Una vez terminada esta etapa se procede a la comparación de los modelos, tanto de Elman, Jordán y los modelos propuestos por Christian Gutiérrez en su Proyecto de tesis basado en Topologías Williams Zipser, con algoritmos de activación Sigmoidea y RBF.

TABLA 5-11 – COMPARATIVA DE MODELOS PROPUESTOS

Modelo	RMSE	MAPE	R^2	PAR	SEG.
Sigmoidea	0,027359825	3,57636819	0,95640655	170	209,35
RBF	0,024971784	3,38151901	0,96687528	196	470,5
Elman	0,073590896	3,22689916	0,60153334	253	957,5
Jordán	0,033632127	1,42027983	0,91446146	139	1194,55

La tabla anterior muestra los resultados de los distintos modelos en sus valores máximos, es con estos resultados que se analizara en el punto 5.4.4 la ganancia de los modelos propuestos. Cabe destacar que el mejor modelo propuesto por Gutiérrez, posee una configuración de 8-IN / 11-OUT versus la mejor red propuesta, la red de Jordan con una configuración de 8-IN / 12-HID / 1-OUT.

5.4.1. Tiempo de Ejecución y Cantidad de Parámetros

El tiempo de ejecución de los algoritmos varía entre un algoritmo y otro, esto debido a topología, cantidad de conexiones, cantidad de parámetros, recursividad, calidad de la programación y capacidad de procesamiento del equipo. En estos casos las redes fueron emuladas en equipos de igual capacidad de procesamiento.

En el caso de las dos primeras redes Sigmoidea y RBF, la recursividad en capas intermedias no existe por lo el procesamiento debido a la no tener recursividad es más rápido que si lo tuviera. En este punto también influye la cantidad de parámetros ya que a mayor cantidad de parámetros para la red esta tiende a tener una mayor demora.

5.4.2. Coeficiente de Determinación R²

Este nos ayuda a interpretar el resultado obtenido de la red como predictor de los modelos propuestos, este valor varía entre 0 y 1. Al acercarse al valor 1, indica una fuerte relación entre los datos de forma que la predicción es más exacta. De lo anterior, según la tabla 5.11 el modelo Sigmoideo y RBF, son los que presentan un valor cercano a 1. Estos corresponden al modelo propuesto por Gutiérrez en su proyecto de título, en cambio en este trabajo el que se acerca más a 1 es el modelo de Jordán.

5.4.3. Interpretación del resultado

Con los resultados obtenidos suponiendo un nivel de captura de 14 millones de toneladas con un r² de 1, tenemos los siguientes resultados:

a) Sigmoidea:	14*0,9564	=13,3896 MT	Error: 14*0,05	= 0,6103 MT.
b) RBF:	14*0,9668	=13,5362 MT	Error.:14*0,0331	= 0,4637 MT.
c) Elman:	14*0,6015	= 8,4214 MT	Error: 14*0,3984	= 5, 5785 MT
d) Jordan:	14*0,9144	=12,8024 MT	Error: 14*0,0855	= 1,1975 MT

Con esto se aprecia que si el predictor seleccionado es Elman de un máximo de extracción de 14 toneladas solo se están extrayendo 8,4214 MT., desaprovechando el recurso, influyendo en la economía del país, en el consumo del recurso, en las fuentes de trabajo, etc. En cambio si se elige un predictor como RBF solo se estaría desaprovechando 0,4637 MT. del recurso, causando un impacto mucho menos fuerte que el caso anterior. Cabe destacar que para poder realizar la comparativa de estos modelos se mantuvieron las generaciones en un valor de 200, aunque las poblaciones utilizadas sean distintas.

5.4.4. Ganancias de los modelos propuestos

De la tabla 5-11 se tomaran los valores de los modelos propuestos en este proyecto de título correspondiente a los modelos de Elman y Jordan, con lo que utilizando la fórmula 4.3. En base a esto se tiene entonces que se calculara la ganancia del modelo de Elman sobre Jordan y a su vez de Jordan sobre Elman.

TABLA 5-122 – COMPARATIVA DE GANANCIA DE MODELOS PROPUESTOS

Modelo	RMSE	MAPE	R ²
ELMAN/SIG	62,8217259	-10,8298714	-58,9947699
ELMAN/RBF	66,0667483	-4,79159225	-60,7351107
ELMAN/JORDAN	54,2985222	55,9862345	-52,0217416
JORDÁN/ELMAN	-118,811305	-127,201647	34,2199353
JORDAN/RBF	25,750209	-138,088223	-5,73165981
JORDAN/SIG	18,6497333	-151,807293	-4,58686252
SIGMOIDEA/ELMAN	-168,974294	9,77161778	37,1048494
SIGMOIDEA/RBF	8,72827586	5,44824161	-1,09458995
SIGMOIDEA/JORDAN	-22,925227	60,2870914	4,38569665
RBF/SIGMOIDEA	-9,56295714	-5,76217905	1,08273841
RBF/ELMAN	-194,69619	4,57249684	37,7858393
RBL/JORDAN	-34,6805138	57,9987625	5,42094943

De la tabla 5-12, se tiene por tanto que el modelo de Jordan tiene una ganancia de un 34,2199353% sobre el modelo de Elman en el coeficiente de determinación, pero en los indicadores RMSE y MAPE tiene significativas pérdidas. Así por contrario el modelo de Elman sobre el modelo de Jordan tiene una pérdida de un 52%. Esta diferencia se produce al existir una gran diferencia entre los modelos.

Si comparamos el modelo de Jordan sobre los propuestos por Gutiérrez, tenemos que al compararlo con el modelo Sigmoideo aun encontramos perdida, pero significativamente menor que en la comparación anterior, esto se debe a la cercanía que tienen entre sus R².

Con esto damos cuenta que al comparar los 4 modelos en general el desempeño de la red RBF es mayor que las restantes configuraciones propuestas, lo que junto con un R² superior nos indica que es el mejor modelo de predicción.

6. Conclusión

Las redes neuronales y los algoritmos genéticos, son usados en sinnúmero de problemas, tanto de ingeniería, economía, biología y otros. Sus aplicaciones crecen día a día y basta únicamente ingenio para acercar el problema que se requiera resolver, a alguna de las combinatorias entre algoritmos genéticos, redes recurrentes y funciones de activación nombradas con anterioridad. La combinatoria estudiada en este documento, fue un perceptron wavelet multicapa recurrente utilizando Elman y Jordan, pudiendo acercar el estudio de la captura de anchoveta, a una predicción basada en esta configuración.

En este trabajo, se refinó el estado del arte, el hacer esto nos permite lograr un desarrollo sólido de los modelos de predicción para la captura mensual de anchovetas, así mediante el uso de algoritmos genéticos y perceptron wavelet recurrente se busca generar una nueva solución a este problema.

Los algoritmos genéticos se utilizaron, para establecer cual sería la codificación de los cromosomas, de manera de representar las redes neuronales que participan en la realización de la predicción. Es por esto que para la modificación de los cromosomas, estos se deben evaluar mediante alguna de las funciones de desempeño propuestos, al igual que la elección de nuevos padres para realizar las mutaciones y generar una nueva población. Gracias a la posibilidad de generar nuevas poblaciones y de mutarlas creamos un universo individuos que nos llevarán a obtener un resultado muy cercano a lo esperado.

Al poder mutar y generar muchas nuevas generaciones, caemos también en una problemática, el encontrar múltiples soluciones locales y no una solución generalizada. Es por esta razón que se utilizan algoritmos de selección, los cuales mediante criterios establecidos serán los que participarán en la selección de individuos para la generación de nuevas poblaciones, estos algoritmos son utilizados para acotar el universo de individuos y de posibles soluciones.

Utilizando por separado algoritmos genéticos y redes neuronales podemos encontrar solución a esta problemática. En este trabajo se creó un algoritmo en donde mezclan, de esta forma, se busca lograr que la convergencia del algoritmo creado, sea más rápida que cada uno por separado. Con este nuevo algoritmo se busca por tanto, encontrar una solución con una convergencia rápida creando una poderosa herramienta.

De los resultados obtenidos se tiene un mejor desempeño en el modelo de Jordán por sobre el de Elman, debido a su topología. Mediante el uso de Jordán y de Elman se establecen modelos predictivos para el volumen mensual de anchovetas y se establecen además ciertos indicadores que nos sirven para calificar el modelo, obteniendo un resultado por sobre lo esperado.

Dentro del análisis de los resultados obtenidos tenemos que el mejor resultado fue para Jordan evolutivo, con un R^2 promedio de 91,44% para un MAPE promedio de 1,42, sobre el modelo de Elman evolutivo, con R^2 de 60,15% y un MAPE de 3,22. Así comparando el mejor resultado obtenido, con el mejor algoritmo propuesto por Gutiérrez, tenemos que su mejor resultado fue con una red RBF, con retroalimentación con un R^2 de 96,68 % y un MAPE de 3,38. Así estableciendo las ganancias sobre los modelos tenemos que la ganancia de la red RBF sobre la red de Jordan en R^2 es un 5,4 % y en el MAPE de un 57,99 %. A su vez la ganancia del modelo de Jordan sobre RBF, tenemos que este tiene una pérdida de un 5,73%, además de tener una gran pérdida en el MAPE con un 138,08 %.

Como se mencionó en la introducción, capítulo 1 y además en el análisis del resultado Capítulo 5 apartado 5, el fin de este informe de título es documentar la generación modelos de pronóstico de captura de anchovetas en millones de toneladas. Se realizaron las pruebas y se acercó el modelo a la realidad.

Este proyecto, no tiene en sus bases la implicancia que puede generar la mayor o menor captura de anchovetas en el lado económico, social sino que en la supervivencia de la especie. Una sobre explotación de anchovetas puede generar la desaparición de esta, como en 1984, en donde la especie estuvo a punto de desaparecer.

7. Referencias

- [1] Shaffer, G., S. Hormazabal, O. Pizarro y S. Salinas. “Seasonal and inter annual variability of currents and temperature over the slope of central Chile”. *J. Geophys. Res.* 104, C12, 29,951-29,961, 1999.
- [2] Shaffer, G., O. Pizarro, L. Djurfeldt, S. Salinas y J. Rutllant. “Circulation and low-frequency variability near the Chilean coast: Remotely forced fluctuations during the 1991-92 El Niño”. *J. Phys. Oceanogr.* 27, 217-235, 1997
- [3] Pizarro, O., G. Shaffer, B. Dewitte y M. Ramos. “Dynamics of seasonal and inter anual variability of the Peru-Chile Undercurrent”. *Geophys. Res. Lett.* 29(12), 10.1029/2002GL 014790, 2002.
- [4] Hormazabal, S., G. Shaffer y O. Pizarro. “Tropical Pacific control of intraseasonal oscillations off Chile by way of oceanic and atmospheric pathways”. *Geophys. Res. Lett.* 29(6), doi: 10.1029/2001GL013481, 2002.
- [5] http://biblioteca.cnice.cl/media/users/3/181868/files/18813/informe_exportaciones_11-01-2010_Rev.pdf
- [6] Tapia, Ricardo. “Las células de la Mente”, Capítulo 3, Las células de la mente FCE 1996.
- [7] McCulloch, Warren y Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”, Vol. 7. *Bulletin of Mathematical Biophysics*, 1943.
- [8] Haykin, Simon. “Neural Networks, a comprehensive foundation”, Second Edition. Pearson Prentice Hall, 1999.
- [9] Demuth, Howard y Mark Beale. *Neural Network Toolbox: User's Guide 3.0*. The MathWorks Inc, 1998.
- [10] Hornik, K. “Multilayer Feed-forward Networks are Universal Approximators”, Vol. 2, 359-366, 1989.

- [11] M. I. Jordan. "Attractor dynamics and parallelism in a connectionist sequential machine", 112-117, 1990.
- [12] C. T. Lin y C. S. G. Lee. "Neural fuzzy systems: a Neuro-fuzzy synergism to intelligent systems". Prentice Hall PTR, Upper Saddle River, N.J. 07458, 1996.
- [13] M. I. Jordan. In R. A. Wilson and F. C. Keil (Eds.). Recurrent networks. The MIT Encyclopedia of the Cognitive Sciences, Cambridge, MA: MIT Press, 1999.
- [14] Funahashi, K., 1989. On the approximate realization of continuous mappings by neural networks. Neural Networks, 2, 183-192.
- [15] Alarcon Aquino, V., Anomaly Detection and Prediction in communication Networks Using Wavelet Transforms, PhD Thesis, Imperial College London, University of London, 2003.
- [16] D. T. Pham y L. Xing. "Neural networks for identification, prediction and control". Springer-Verlag, London, Great Britain, 1995.
- [17] Miller, Brad L. y David E. Goldberg. "Genetic Algorithms, Tournament Selection, and the Effects of Noise". Vol. 9. Complex Systems, 1995.
- [18] David E. Goldberg. Computer-aided gas pipeline operation using genetic algorithms and rule learning, PhD thesis. University of Michigan. Ann Arbor, MI, 1983.
- [19] Mitchell, Melanie. An Introduction to Genetic Algorithms. The MIT Press, 1999.
- [20] DeJong, K. A. (1975). An Analysis of the Behavior a Class of Genetic Adaptive Systems. PhD thesis, University of Michigan.
- [21] Kecman, Vojislav. "Learning and soft computing: support vector machines, neural networks, and fuzzy logic models". The MIT Press, 2001.
- [22] Gutiérrez, Christian. Proyecto de tesis, "Predicción de captura de Anchovetas utilizando Redes Neuronales Evolutivas Sigmoidales y RBF con Algoritmos Genéticos", Pontificia Universidad Católica de Valparaíso, 2009.