UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

"PRONÓSTICO DE UNA SERIE TEMPORAL USANDO REDES NEURONALES"

TESIS

PARA OBTENER EL TÍTULO DE:

LICENCIADO EN MATEMÁTICAS APLICADAS

PRESENTA:

WILLIAMS GÓMEZ LÓPEZ

DIRECTOR:

M.C. JOSÉ DEL CARMEN JIMÉNEZ HERNÁNDEZ

CO-DIRECTOR:

DR. FELIPE DE JESÚS TRUJILLO ROMERO

Huajuapan de León, Oaxaca **Agosto de 2010.**

Pronóstico de una serie temporal usando redes neuronales

Williams Gómez López

Agosto de 2010

A mis padres, Arturo y María,
y a mis hermanos, Wilbher y Daniel.

Índice general

Prólogo

El pronosticar algún fenómeno repercute en la toma de decisiones de una empresa y en la planificación de recursos para una mayor y eficiente producción. Además de lo anterior el conocer que suceso va a pasar en el futuro permite tomar ciertas medidas preventivas.

El realizar pronóstico conlleva desde entender la técnica que se esta utilizando hasta saber los posibles costos que se generará al realizar esto. Claro esta que los datos que se coleccionen deberán ser fiables ya que al momento de la construcción de algún modelo, la predicción deba ser lo mas realista posible y además ser confiable.

Por lo cual en este trabajo de tesis el objetivo principal es realizar la predicción y la comparación de estos para un conjunto de datos los cuales corresponden a las demandas máximas mensuales de energía eléctrica proporcionados por la Comisión Federal de Electricidad. Son dos los enfoques estudiados para lograr tal objetivo, la metodología de Box y Jenkins para series de tiempo y el uso de las redes neuronales artificiales. Esta última ha brindado soluciones a problemas en donde los datos presentan algún ruido o que se tengan datos faltantes a la hora de resolver dicho problema. A continuación se describe de manera general los puntos importantes de cada Capítulo que esta conformada la tesis.

En el primer Capítulo se revisan algunos puntos en la realización de pronósticos además de discutir las diferentes técnicas que existen para llevarlo a cabo.

En el Capítulo dos se revisa la teoría de series de tiempo como son procesos estocásticos, la definición formal de series de tiempo, los principales modelos, los criterios utilizados para el buen ajuste de estos y finalmente el pronóstico.

Los modelos de redes neuronales, su estructura, las diferentes topologías y principalmente la revisión de las redes neuronales multicapa además del algoritmo *backpropagation* y sus diferentes variantes se estudian en el Capítulo tres.

El pronóstico para los datos utilizando los dos enfoques y la comparación se realizan en el Capítulo cuatro.

Finalmente en el Capítulo cinco se muestran las conclusiones obtenidas en este trabajo.

Como punto final, para el análisis de los datos se utilizó el software estadístico `R project` [16] el cual se distribuye gratuitamente bajo los términos de la *GNU General Public Licence*

y MATLAB© que es un software la simulación matricial, específicamente el *toolbox* de redes neuronales [6].

Capítulo 1
Preliminares

Toda institución, empresa o el gobierno tienen que hacer planes para el futuro si desea progresar o utilizar adecuadamente sus recursos financieros, para ello se requiere conocer el comportamiento futuro de ciertos fenómenos con el fin de planificar o prevenir. Una planificación exige prever los sucesos del futuro que probablemente vayan a ocurrir. La previsión a su vez se suele basar en lo que ha ocurrido en el pasado. Una técnica estadística para hacer inferencia sobre el futuro con base en lo que ha ocurrido en el pasado es el análisis de series de tiempo, es decir, uno de los problemas que intenta resolver ésta es la predicción. Existen muchas áreas de la ciencia en donde se presenta este problema, por ejemplo, la economía, física-química, electricidad, telecomunicaciones, etc. Como alternativa para resolver este problema, sin argumentos puramente estadísticos, se encuentran las redes neuronales artificiales. Son una nueva herramienta de computación capaz de manejar imprecisiones e incertidumbre, los cuales aparecen cuando se trata de resolver problemas relacionados con el mundo real, ofreciendo soluciones robustas y de fácil implementación.

1.1. Introducción a la predicción

La acción de predecir hechos o condiciones futuras se le denomina ***pronosticar***, en otras palabras, realizar un ***pronóstico*** es obtener noción acerca de hechos futuros. Por ejemplo:

- En la producción de una empresa es necesario conocer la demanda de un artículo en el siguiente periodo. Esta predicción se realizan en periodos específicos y permiten a la empresa realizar la planificación de la producción, el mantenimiento del inventario y además de conocer la cantidad de materia prima que serán necesarios para poder cubrir la demanda del artículo en el siguiente periodo.

- En el control de procesos de una industria es importante el pronosticar el comportamiento que va a tener el proceso. Por ejemplo, un proceso industrial podría comenzar a producir cantidades de productos defectuosos cuando el proceso opera en tiempo extra. Si fuese posible predecir este comportamiento, entonces se podría realizar una inspección y el mantenimiento preventivo necesario de modo que el número de productos defectuosos sean mínimos.

Con el objeto de realizar una predicción confiable se debe de analizar los datos disponibles, esto para identificar algún patrón que pueda utilizarse en la descripción de los mismos. El patrón obtenido se extrapola, es decir, se amplia hacia el futuro y con esto obtener un pronóstico, aclarando que el patrón identificado anteriormente no sufra algún cambio en el futuro. Si el patrón que se identifico sufre algún cambio, entonces la técnica de predicción obtenida no generará un buen pronóstico, sin embargo se puede realizar cambios apropiados en el patrón de modo tal que se evite la inexactitud en el pronóstico.

1.1.1. Métodos de pronósticos

Existen una gran gama de métodos para realizar pronósticos, los cuales podemos dividirlos en dos tipos básicos: los métodos ***cuantitativos*** y los métodos ***cualitativos***.

Los ***métodos cualitativos*** requieren la opinión de expertos para realizar una predicción de forma subjetiva y obtener así hechos futuros. Este tipo de método se utiliza cuando los datos históricos no están disponibles o no son lo suficiente para realizar el modelo de predicción. Por ejemplo, para pronosticar las ventas de un producto nuevo en el mercado la compañía debe de confiar en la opinión de expertos para la construcción de un modelo para la predicción.

Además de proporcionar modelos para el pronóstico, los métodos cualitativos son utilizados para describir el comportamiento de las variaciones del patrón que fue utilizado para el desarrollo del modelo.

Existen varias técnicas cualitativas para realizar un pronóstico, una de estas técnicas requiere un ***ajuste de curva subjetiva*** (Véase [2]). Por ejemplo, una empresa podría estar interesada en el pronóstico de las ventas de un artículo nuevo que sacará al mercado, para esto la empresa recurre a la opinión de sus expertos de ventas y mercadotecnia para la construcción subjetiva de una curva S, la cual se construye al aplicar la experiencia que tiene la empresa en la venta de otros artículos similares con la finalidad de realizar la predicción del crecimiento de las ventas y también del tiempo que durará este crecimiento y cuando se *estabilizarán* las ventas en el mercado.

Uno de los problemas que tiene este tipo de técnica es el elegir la forma que tendrá la curva S incluyendo además la gran dificultad de la construcción de la curva, así el experto requerirá de una gran experiencia y criterio en la elaboración de la curva.

Otro método cualitativo bastante conocido para el pronóstico es el ***método Delphi*** (Véase [2]). Funciona de la siguiente manera, suponga que un equipo de expertos (reconocidos en el campo de interés) plantean predicciones relacionadas acerca de un hecho especifico, haciendo el supuesto de que la combinación del conocimiento de cada uno de ellos generará una predicción más fiable que la que nos brindaría cada experto por separado. Así mismo, se evita el problema que existiría a la hora de obtener el pronóstico de todo el equipo, puesto que habría ideas que muchos de los expertos no estarían de acuerdo y además las discusiones estarían bajo un sólo experto o un pequeño grupo de ellos. Por

este motivo se separa a cada experto y se obtiene el pronóstico de cada uno de ellos, el cual será entregado a un coordinador que reorganizará los pronósticos y que posteriormente este nuevo pronóstico sea revisado por cada experto. En lo anterior, cada uno de ellos evaluará por separado este nuevo pronóstico esperando que después de varias rondas se obtenga un buen pronóstico. Delphi no trata de llegar a una unanimidad de los pronósticos de cada uno de los expertos, por el contrario, permite realizar diferencias justificadas de las opiniones de cada uno de ellos.

Una tercera técnica cualitativa para hacer pronósticos es la que realiza una ***comparación de técnicas*** que son independientes del tiempo, esta técnica junto con las que se mencionaron anteriormente reciben el nombre de ***métodos de pronósticos que requieren una opinión***.

Por otra parte las ***técnicas cuantitativas de predicción*** consisten en encontrar algún patrón en los datos disponibles para poder proyectarlo al futuro, es decir estos métodos requieren de un análisis de la información para poder efectuar la predicción de la variable que sea de interés. Estos métodos son muy utilizados y se clasifican en dos tipos de modelos, ***modelos univariados*** y ***modelos causales***.

Los modelos univariables analizan los datos con el objeto de identificar algún patrón, suponiendo que dicho patrón no se vea afectado en el transcurso del tiempo, se extrapola para poder generar la predicción. Por ejemplo, se puede usar un modelo univariable para predecir las ventas que una compañía espera tener al usar alguna estrategia de mercado. Sin embargo, dicho modelo no podrá predecir los cambios de las ventas esto por el resultado del incremento del precio, gastos excesivos de publicidad o el cambio de la campaña publicitaria, entre otras causas que repercutan en el cambio del patrón obtenido de los datos.

Los modelos causales requieren identificar variables que se relacionan con la variable a predecir. Una vez que se hayan encontrado estas variables, se desarrolla un modelo estadístico que describirá la relación que existe entre estas variables con la variable que se desea pronosticar. Como ejemplo, las ventas de un producto puede estar relacionado con los gastos de publicidad, los precios de los competidores del mismo producto además del precio del producto que la compañía esta ofreciendo en el mercado. En este caso, la variable venta recibe el nombre de ***variable dependiente*** mientras que las demás variables reciben el nombre de ***variables independientes***.

En los negocios, los modelos causales ofrecen ventajas al momento de realizar algún pronóstico, ya que evalúa el impacto que tendrán sus estrategias de mercado en el mundo financiero. Sin embargo, existen varias desventajas al aplicar este modelo, como son

- Dificultad en el desarrollo.
- Requieren de datos anteriores de cada variable que se incluya en el modelo.
- Requieren de una gran habilidad por parte del experto para poder realizar el pronóstico.

1.1.2. Seleccionando una técnica de pronóstico

Antes de elegir una técnica de pronóstico, es necesario considerar los siguientes factores:

1. Periodo.
2. Patrón de los datos.
3. Costo del pronóstico.
4. Exactitud deseada.
5. Disponibilidad de información.
6. Facilidad de operar y entender.

La duración del periodo contribuye en la elección de la técnica de predicción, donde este se clasifica en:

- *Inmediato*, el cual contemplan un plazo de menos de un mes.
- *Corto Plazo*, que va desde un mes a tres meses.
- *Medio Plazo*, transcurre más de tres meses a menos de dos años.
- *Largo Plazo*, este periodo va de dos años o más.

Los costos llevados a cabo en la realización del pronóstico son considerables (costos por desarrollar el modelo y el costo de operación del mismo), ya que algunos de los métodos de pronósticos son muy sencillos de operar, sin embargo otros son complejos, así el grado de complejidad puede tener una influencia definitiva en el costo total del pronóstico.

Otro punto a considerar es la exactitud deseada en el pronóstico. Este dependerá del ámbito en donde se desea realizar el pronóstico pues bien, un error de predicción de hasta un 20 % podría ser a veces aceptable pero otras veces un pronóstico que tenga un error de 1 % podría ser no muy conveniente para diversas situaciones en los cuales la precisión debe ser casi exacta.

Por otra parte distintas técnicas de pronósticos requieren diferentes cantidades de datos, con ello no solo la disponibilidad de los datos es importante si no que también la exactitud y la puntualidad de los datos con que se cuentan, puesto que si los datos son obsoletos o inexactos originaran predicciones inexactas además de esto se necesita algún procedimiento para poder recabar los datos eficazmente.

Como último punto el entendimiento del método que se esta aplicando es importante ya que al no saber y comprender la técnica que se desea aplicar, no se tendrá confianza en los resultados del modelo y por lo tanto éstos resultados no se tendrán en cuenta, como por ejemplo la toma de decisiones de una empresa.

1.2. Pronósticos y series de tiempo

Al conjunto de observaciones en donde cada valor queda determinado de manera cronológica se le denomina ***serie de tiempo***. Ejemplos de esto se presentan en una variedad de campos, desde la economía: en donde podemos involucrar los precios de las acciones que ocurren diariamente en la bolsa de valores, el total de exportaciones mensuales de una empresa; en la medicina: un epidemiólogo puede estar interesado en el número de casos de gripe observado en un determinado periodo; la geología: los registros sísmicos pueden ayudar en los trazos de fallas o la distinción entre un terremoto o una explosión nuclear.

El análisis de una serie de tiempo se realiza con el objetivo de emplear modelos ya establecidos que faciliten la descripción de los datos proporcionados previamente. La serie está compuesta de varias componentes las cuales se enuncian a continuación:

1. ***Tendencia***

   Es el componente de largo plazo que representa el crecimiento o declinación de la serie. En términos intuitivos, la tendencia caracteriza el patrón gradual y consistente de las variaciones de la serie, que es consecuencia de "fuerzas persistentes" que afectan el crecimiento o la reducción de la serie.

2. ***Ciclo***

   Es la fluctuación en forma de onda alrededor de la tendencia. Una de las fluctuaciones cíclicas más comunes en series de tiempo son las llamadas ciclo económico. La cual esta representado por fluctuaciones ocasionadas por periodos recurrentes de prosperidad alternando con recesión. Sin embargo, dichas fluctuaciones no necesitan ser causadas por cambios en los factores económicos, como por ejemplo; en la producción agrícola donde la fluctuación cíclica puede estar ocasionada por los cambios climáticos.

3. ***Variación Estacional***

   Son patrones periódicos que se repiten año tras año, factores como el clima y las costumbres ocasionan estos tipos de patrones.

4. ***Fluctuaciones irregulares*** o ***irregularidad***

   Son movimientos erráticos que siguen un patrón indefinido o irregular. Estos movimientos representan lo que queda de la serie después de haber restado las demás componentes. Muchas de las fluctuaciones irregulares son causadas por hechos inusuales que no se pueden predecir, como son los sismos, huracanes, guerras, entre otros.

No todas las componentes de una serie de tiempo se presentan solas, sino que podrá ser la combinación de dos o más componentes mencionados anteriormente. Por ejemplo, un modelo de pronóstico que pueda ser utilizado para predecir una serie de tiempo que sea

caracterizada por la tendencia no será apropiado para predecir series caracterizadas por una combinación de tendencia y variación estacional. De aquí, es necesario obtener un modelo de predicción apropiado para el patrón de los datos disponibles.

Una vez que se haya obtenido un modelo adecuado, entonces se estiman las componentes de la serie de tiempo, los cuales serán los parámetros del modelo para después ocupar las estimaciones y realizar así un pronóstico. Los modelos de series de tiempo suponen que la serie puede estar expresada como ***producto*** o como ***suma*** de las cuatro componentes de dicha serie.

De las técnicas cuantitativas para series de tiempo univariables se pueden mencionar

1. *Regresión de serie de tiempo.* Técnica que relaciona la variable dependiente con funciones de tiempo que describan la componente de tendencia y la componente de variación estacional.

2. *Métodos de descomposición.* Como su nombre lo indica, descompone la serie de tiempo en sus cuatro componentes con el objetivo de describir cada una de ellas por separado logrando así describir y predecir en conjunto la serie de tiempo.

3. *Suavizamiento exponencial.* El objetivo es filtrar o suavizar la serie para tener una mejor idea del comportamiento de la tendencia y por tanto tener un pronóstico más confiable.

4. *Metodología de Box-Jenkins.* Proporciona una colección más extensa de modelos de predicción además de ser un procedimiento más sistemático para ayudar a identificar el modelo adecuado.

1.2.1. Error del pronóstico

Al realizar algún pronóstico, se considera tener un error en la predicción. Recuerde que la componente irregular de una serie de tiempo está representada por fluctuaciones inexplicables o impredecibles en los datos, por esto se espera algún error al realizar el pronóstico puesto que si la componente irregular es notable los pronósticos serán inexactos. Por el contrario, si dicha componente no es notoria, la determinación de las demás componentes permitirá un pronóstico confiable.

Hay que destacar aquí, que no solamente la componente irregular brinda *incertidumbre* al momento de realizar el pronóstico, si no que también existe incertidumbre al momento de predecir las demás componentes de la serie, motivo por el cual no se pueda realizar con exactitud un pronóstico en la práctica.

De lo anterior, se dice que con errores de predicción muy grandes, la componente irregular es tan grande que ninguna técnica será capaz de generar un pronóstico exacto o que la técnica utilizada no es la adecuada para poder predecir la tendencia, la variación estacional o el componente cíclico.

Una de las preocupaciones al realizar un pronóstico, es el medir el error que se comete al tratar de predecir una variable. Supongase que se denota el valor real de la variable de interés en el tiempo t mediante y_t y el valor que se predijo con el modelo por $\hat{y}_t$, entonces el ***error de pronóstico*** (denotado por e_t) para un valor particular $\hat{y}_t$ será, $e_t = y_t - \hat{y}_t$.

Así, para saber si una técnica de predicción es adecuada para el patrón obtenido de la serie, se realiza un análisis en el error de predicción. Por ejemplo, si una técnica de predicción describe completamente la tendencia, la variación estacional o el componente cíclico que están presentes en la serie, entonces los errores de pronósticos que se obtengan reflejará solo la componente de irregularidad, es decir, los errores del pronóstico deberán ser aleatorios.

Cuando la técnica de predicción no concuerda con el patrón de los datos, entonces los errores de pronóstico manifiestan un patrón con respecto al tiempo. Por ejemplo, los errores manifestarían una tendencia hacia arriba o un patrón estacional o más aún, un patrón cíclico los cuales no explicarían a la perfección el patrón de los datos, esto es, el modelo dará predicciones inexactas.

Si los errores de pronóstico en el tiempo indican que la metodología usada para el pronóstico es la adecuada, entonces sería importante medir la magnitud de los errores de modo que permita determinar si puede realizarse un pronóstico exacto. Para esto existen diversas maneras de realizarla, como por ejemplo, la ***desviación absoluta media*** (DAM) y el ***error cuadrático medio*** (ECM).

La diferencia básica de estas dos medidas es que el ECM *penaliza* más a la técnica para pronosticar que la DAM en los errores grandes que en los errores pequeños. Se menciona por ejemplo, un error de 2 genera un error cuadrático de 4, sin embargo, un error de 4, da un error cuadrático de 16.

Estas medidas se pueden usar de dos formas distintas, en primer lugar puede ayudar en el proceso de elección de un modelo de pronóstico y una manera común de realizar esta elección es realizando una simulación de los datos anteriores. En el proceso de simulación se supone que no se conocen los valores de los datos anteriores. Se usa cada modelo de pronóstico para generar predicciones de los datos anteriores para que posteriormente se realice una comparación de estas predicciones con los datos reales. Se realiza una medición de los errores de pronóstico y como segunda forma, estas mediciones se pueden emplear para el monitoreo de un sistema de pronósticos esto con el objeto de detectar alguna anomalía en el sistema. Por ejemplo, supongase que el patrón de los datos que se obtuvo y que ha estado presente durante un periodo largo cambia repentinamente, esto provocaría que el modelo de pronóstico que se esta usando podría dar predicciones inexacta en la variable de interés, de aquí que se debe de anticipar este cambio antes de que el pronóstico se vuelva inexacto. De lo anterior se incorporan la DAM y el ECM con el fin de que estos indiquen cuándo los errores se vuelven demasiado grandes.

1.3. Pronóstico y Redes Neuronales Artificiales

La teoría de las ***Redes Neuronales Artificiales*** (RNA), ha brindado una alternativa a la computación clásica para aquellos problemas, en los cuales los métodos tradicionales no han entregado resultados muy convincentes además de que éstos modelos están inspirados en tratar de emular el comportamiento inteligente de sistemas biológicos. Las aplicaciones más exitosas de las RNA son:

1. Procesamiento de imágenes y de voz.
2. Reconocimiento de patrones.
3. Planeamiento.
4. Interfaces adaptativas para sistemas Hombre-Máquina.
5. Predicción.
6. Control y optimización.
7. Filtrado de señales.

El objetivo principal de las RNA es la emulación abstracta de los sistemas nerviosos biológicos, los cuales están formados por un conjunto de unidades llamadas *neuronas* o *nodos* interconectados cada uno de ellos. El primer modelo de red neuronal fue propuesto en 1943 por el neurofisiólogo Warren McCulloch y el matemático Walter Pitts quienes habían modelado una red neuronal simple mediante circuitos eléctricos (Véase [11]).

El desarrollo de una red neuronal se pueden realizar en periodos de tiempos razonables y realizar tareas concretas mucho mejor que otros enfoques. Existen diversos tipos de redes neuronales cada uno con una aplicación particular más apropiada. Uno de los modelos más utilizados es el modelo del *perceptrón*, el cual proporciona buenos resultados al resolver problemas en el ámbito financiero, tal es el caso de la *prediccion* de eventos.

La neurona artificial como unidad independiente no es muy eficaz para el tratamiento de la información, de aquí que es importante la implementación de redes *multicapa*. Cada neurona está caracterizada por un valor numérico denominado ***estado de activación*** y asociada a cada unidad, existe una ***función de transmisión*** que transforma el estado actual de activación en una ***señal de salida***. Esta señal es enviada a través de los canales de comunicación a otras unidades de la red; en estos canales la señal es modificada según sea el ***peso*** o ***sinapsis*** el cual esta asociada a cada uno de estos según una determinada regla.

Una ***función de activación*** determinará el nuevo estado de activación de la neurona, para esto, deberá estar considerando todas las señales de entradas que han sido modificadas por los pesos de cada conexión de neuronas que están conectadas con esta, es decir, la función solo dependerá de los datos de entrada en la red neuronal, (Véase [11]).

La forma en que las neuronas se distribuyen dentro de la red neuronal multicapa es mediante ***niveles*** o ***capas*** las cuales están determinadas por un número determinado de neuronas las cuales se clasifican en tres tipos:

- ***Entrada***

  Esta capa recibe la información de fuentes externas de la red.

- ***Oculta***

  En esta capa es donde se procesa la información de la capa exterior para que posteriormente sea enviada a la capa de salida. Estas neuronas pueden ser conectadas de diferente manera incluyendo el número de neuronas que formen la capa, determinará una gran variedad de topologías de la red.

- ***Salida***

  Estas neuronas transfieren la información hacia el "exterior".

Las RNA se han empleado en la resolución de diversos problemas en donde destaca los problemas financieros cuya aplicación principal es la predicción. Aunque los datos a analizar estén incompletos o los datos presenten cierta dependencia de otras variables para su obtención, los resultados obtenidos al usar RNA son satisfactorios.

La aplicación de las RNA se divide en dos categorías: ***clasificación y modelado***. En la primera se discrimina las observaciones por características comunes en diferentes grupos, como por ejemplo, predicción de fallas corporativas, la clasificación de bonos, entre otros, mientras que el segundo consiste en simular el comportamiento de una entidad o variable basado en observaciones previas de los datos, por ejemplo, predicción de las fluctuaciones de los precios de las acciones o del tipo de cambio, (Véase [17]).

Además de las aplicaciones que se mencionó anteriormente, existen otras aplicaciones de las RNA, como compresión de imágenes, reconocimiento de voz, aplicaciones de apoyo a la medicina[1] y todo tipo de aplicaciones que necesiten el análisis de grandes cantidades de datos.

[1]Tal es el caso del problema de clasificación de diagnóstico de la diabetes Tipo II (Véase[12]).

Capítulo 2

Series de tiempo

2.1. Procesos Estocásticos

El propósito del análisis de una serie de tiempo es entender o modelar mecanismos estocásticos que se dan lugar en una serie observada y además, predecir o pronosticar valores futuros de la serie basándose en la historia de ésta y posiblemente otros factores relacionados con la serie. Para poder estudiar de manera adecuada una serie de tiempo, es necesario tener la noción acerca de lo que es un *proceso estocástico*, cuya definición se menciona a continuación.

Definición 2.1.1. Un ***proceso estocástico*** es una familia de variables aleatorias $\{X_t : t \in T\}$ las cuales están asociadas a un conjunto de índices T, comúnmente interpretado como el tiempo, de tal forma que a cada elemento del conjunto le corresponda una y sólo una variable aleatoria. Al conjunto de todos los posibles valores que la variable aleatoria pueda tomar se le llama *espacio de estados* el cual se denota por S.

Si el conjunto de índices que se está considerando es un intervalo ya sea cerrado o abierto, se dirá que el proceso estocástico es en tiempo continuo, por otro lado, si el conjunto de índices es un conjunto finito o infinito numerable el proceso será llamado proceso estocástico en tiempo discreto. Además de la clasificación que se ha dado anteriormente, existe otro tipo de división en donde los procesos estocásticos se clasifican de acuerdo a la cardinalidad del espacio de estados, así se tendrán procesos estocásticos en tiempo continuos con espacio de estado discreto o continuo y proceso estocástico en tiempo discreto con espacio de estado discreto o continuo.

Una manera para poder describir a un proceso estocástico, es conociendo su distribución de probabilidad conjunta, sin embargo, en la práctica suele ser muy complicado y hasta imposible conocerse. Otra forma es conociendo los ***momentos*** del proceso, particularmente el primer y segundo momento, es decir, las ***medias*** y las ***varianzas*** de las variables involucradas en el proceso estocástico. Por otra parte, la varianza no es lo suficiente para poder especificar el segundo momento del proceso, motivo por el cual es necesario definir funciones que ayuden a caracterizar ciertas propiedades del proceso. Estas funciones son la *función media* y la *función de autocorrelación* las cuales se definen a continuación.

Definición 2.1.2. Sea $\{X_t : t \in T\}$ un proceso estocástico.

1. La ***función media*** se define como:

$$\mu_X(t) = \mathrm{E}[X_t] \quad \text{para} \quad t \in T.$$

2. Si para cada $t \in T$ se tiene que $\mathrm{Var}(X_t) < \infty$, entonces la función ***Función autocovarianza***, denotada por $\gamma_X(t,s)$ se define como:

$$\begin{aligned}\gamma_X(t,s) &= \mathrm{Cov}(X_t, X_s) \\ &= \mathrm{E}[(X_t - \mu_X(t))(X_s - \mu_X(s))] \quad \text{para} \quad t,s \in T.\end{aligned}$$

3. La ***función autocorrelación***, denotada por $\rho_X(t,s)$, esta definida por:

$$\begin{aligned}\rho_X(t,s) &= \mathrm{Corr}(X_t, X_s) \quad \text{para} \quad t,s \in T \\ &= \frac{\mathrm{Cov}(X_t, X_s)}{\sqrt{\mathrm{Var}(X_t)\,\mathrm{Var}(X_s)}} \\ &= \frac{\gamma_X(t,s)}{\sqrt{\gamma_X(t,t)\gamma_X(s,s)}}.\end{aligned}$$

Para realizar inferencia estadística acerca de un proceso estocástico, se deben de realizar ciertas suposiciones acerca de la estructura o comportamiento del proceso, y una de las más importante es la *estacionalidad*.

Definición 2.1.3. Un proceso estocástico es ***estrictamente estacionario*** si la distribución de probabilidad conjunta de cualquier colección de variables $\{X_{t_1}, X_{t_2}, \ldots, X_{t_k}\}$ es la misma que de la colección de variables con un desplazamiento h, $\{X_{t_1+h}, X_{t_2+h}, \ldots, X_{t_k+h}\}$, es decir

$$P\{X_{t_1} \leq x_1, \ldots, X_{t_k} \leq x_k\} = P\{X_{t_1+h} \leq x_1, \ldots, X_{t_k+h} \leq x_k\},$$

para cualquier k, t, todo número $x_1, \ldots, x_k$ y todo desplazamiento $h \in \mathbb{Z}$.

Aunque en la práctica, suele definirse la estacionalidad de un proceso estocástico un poco menos restringida.

Definición 2.1.4. Un proceso estocástico $\{X_t : t \in T\}$ se dice que es ***débilmente estacionario*** o de ***estacionalidad de segundo orden*** si

1. $\mathrm{E}[X_t] = \mu$ para todo $t \in T$.

2. $\gamma_X(t+h,t)$ es independiente de cualquier valor t para cada valor h.

De esta última definición, se tiene que todo proceso que cumpla que su primer y segundo momento no dependan del tiempo, se les llamarán *procesos estacionarios de segundo orden.*

Si se tiene que el segundo momento central de un proceso estocástico *estrictamente estacionario* es finito, es decir, $\mathrm{E}[X_t^2] < \infty$ entonces dicho proceso es *débilmente estacionario.*

Un proceso estocástico importante es aquel en donde las variables aleatorias son independientes e idénticamente distribuidas con media cero y varianza σ^2, tal colección de variables aleatorias, $\{X_t : t \in T\}$, es conocido como ruido independiente e idénticamente distribuido, el cual estará denotado por $X_t \sim \mathrm{IID}(0,\sigma^2)$. Dicho proceso tiene un papel importante para poder construir modelos que permitan describir series de tiempo. Por otra parte, si además se tiene que las variables no están correlacionadas entonces $\{X_t : t \in T\}$ es llamado ***ruido blanco*** que será denotado mediante $X_t \sim \mathrm{WN}(0,\sigma^2)$. A continuación se define lo que es una *serie de tiempo.*

Definición 2.1.5. Una ***serie de tiempo*** es una sucesión de observaciones generadas por un proceso estocástico $\{X_t : t \in T\}$ donde el conjunto de índice T esta en función del tiempo.

De esta forma, un modelo de serie de tiempo para un conjunto de datos observados $\{x_t\}$, es la especificación de la distribución de probabilidad conjunta de una sucesión de variables aleatorias X_t, para las cuales $\{x_t\}$ se toma como una realización del proceso.

2.2. Modelos de series de tiempo

Los métodos tradicionales para el análisis de series de tiempo, se refieren principalmente a la descomposición de la serie en sus fuentes de variaciones tales como tendencia, variación estacional, variación cíclica y fluctuaciones irregulares. Este enfoque no siempre es bueno (Véase [2]) sin embargo dicho método se tiene cuando las fuentes de variaciones de la serie sólo las rigen la tendencia y/o la variación estacional.

Modelo de tendencia

Un modelo en el análisis de series es el que hace el supuesto de que la componente de variación estacional esta ausente, es decir, la serie de tiempo tiene una estructura simple de tendencia m_t, el cual puede ser expresado de la siguiente manera:

Definición 2.2.1 (Modelo no estacional con tendencia)**.**

$$X_t = m_t + \varepsilon_t, \quad \text{para} \quad t = 1,\ldots,n,$$

donde $\mathrm{E}[\varepsilon_t] = 0$.

Si $\mathrm{E}[\varepsilon_t] \neq 0$ entonces se hace un reemplazo para m_t y ε_t mediante $m_t + \mathrm{E}[\varepsilon_t]$ y $\varepsilon_t - \mathrm{E}[\varepsilon_t]$ respectivamente. Es usual el suponer que $m_t = \alpha + \beta t$, el cual representa un modelo lineal de tendencia, y puede ser estimado mediante mínimos cuadrados, en donde se busca minimizar $\sum_t (x_t - m_t)^2$.

Modelo de estacionalidad

Otro modelo de serie de tiempo, es suponer que la tendencia no esta presente en la serie y sólo quede expresada mediante la componente de variación estacional s_t.

Definición 2.2.2 (Modelo con variación estacional sin tendencia)**.**

$$X_t = s_t + \varepsilon_t, \quad \text{para} \quad t = 1, \ldots, n.$$

Modelos de estacionalidad y tendencia

En las aplicaciones es posible que un modelo de tendencia o de estacionalidad no ayude a describir el comportamiento de los datos que se tienen, es por ello que existen modelos que tratan de explicar este comportamiento mediante la combinación de los modelos que se describieron anteriormente. Son tres tipos de modelos que son empleados comúnmente, los cuales a continuación se definen.

Definición 2.2.3. Modelos de descomposición.
Sea $\{X_t : t \in T\}$ una serie de tiempo, m_t, s_t, ε_t las componentes de tendencia, variación estacional y la componente de fluctuación o ruido, respectivamente. Se define los siguientes modelos:

1. **Modelo aditivo o clásico**. Este modelo expresa a la serie de tiempo como la suma de las componente de tendencia, componente estacional y la componente de fluctuación, es decir,
$$X_t = m_t + s_t + \varepsilon_t, \quad \text{para} \quad t = 1, 2, \ldots, n \ ,$$
donde $\mathrm{E}[\varepsilon_t] = 0$, $s_{t+d} = s_t$ y además $\sum_{j=1}^{d} s_j = 0$.

2. **Modelo multiplicativo**. Modelo que expresa a la serie como el producto de sus respectivas componentes, es decir,
$$X_t = m_t \ s_t \ \varepsilon_t.$$

3. **Modelo mixto**. Modelo que relaciona al modelo aditivo y al modelo multiplicativo definidas anteriormente, es decir,
$$X_t = m_t \ s_t + \varepsilon_t.$$

Los modelos anteriormente mencionados, se emplean dependiendo de las características de las componentes de la serie. Si la serie a estudiar presenta variación estacional creciente o decreciente es apropiado el uso del modelo de descomposición multiplicativo, por otra parte, si la variación estacional es constante es útil el uso del modelo de descomposición aditiva (Véase [2]).

2.3. Series de tiempo estacionarias

Definición 2.3.1. Sea $\{X_t : t \in T\}$ una serie de tiempo. Se dice que $\{X_t : t \in T\}$ es ***estrictamente estacionaria*** si

$$(X_1, X_2, \ldots, X_n)' \stackrel{d}{=} (X_{1+h}, X_{2+h}, \ldots, X_{n+h})',$$

para todo entero h y $n \geq 1$. (La notación aquí utilizada $\stackrel{d}{=}$ indica que los dos conjuntos de variables aleatorias tienen la misma función de distribución de probabilidad). Es decir, la distribución de probabilidad conjunta de un conjunto arbitrario de variables es invariante respecto a cualquier desplazamiento en el tiempo.

Definición 2.3.2. Una serie de tiempo es ***débilmente estacionaria*** o de ***segundo orden*** si tiene varianza finita y además:

1. La media de cada X_t es constante, es decir, $\mathrm{E}[X_t] = \mu$ para todo t.
2. La función covarianza, $\gamma_X(r, s)$, depende de r y s sólo mediante su diferencia $|r - s|$, es decir $\gamma_X(r, s) = \gamma_X(r + t, s + t)$.

De aquí en adelante, cuando se haga mención de una serie de tiempo estacionaria, se estará refiriendo a una serie de tiempo de segundo orden, salvo que se diga lo contrario.

Si la serie de tiempo es estacionaria, entonces por la Definición 2.3.2 propiedad 2 y tomando el caso particular de $t = -s$ la función de autocovarianza queda de la siguiente manera:

$$\begin{aligned}\gamma_X(r, s) &= \gamma_X(r + t, s + t) \\ &= \gamma_X(r - s, 0) \\ &= \gamma_X(h, 0).\end{aligned}$$

De esto, se tiene que la función de autocovarianza sólo depende de un *lapso* o *retraso* h, que será independiente de cada valor $t \in T$.

Análogamente se tiene que la función de autocorrelación está dado por

$$\begin{aligned}\rho_X(r, s) &= \mathrm{Corr}(X_r, X_s) \\ &= \frac{\gamma_X(r, s)}{\sqrt{\gamma_X(r, r)\gamma_X(s, s)}} \\ &= \frac{\gamma_X(r + t, s + t)}{\sqrt{\gamma_X(r + t, r + t)\gamma_X(s + t, s + t)}} \\ &= \frac{\gamma_X(r - s, 0)}{\sqrt{\gamma_X(r - s, r - s)\gamma_X(0, 0)}} \\ &= \frac{\gamma_X(h, 0)}{\sqrt{\gamma_X(h, h)\gamma_X(0, 0)}}.\end{aligned}$$

Además, se puede observar que $\gamma_X(h,h) = \gamma_X(0,0)$, así se concluye que la función de autocorrelación queda como

$$\rho_X(r,s) = \frac{\gamma_X(h,0)}{\gamma_X(0,0)}.$$

Con el análisis anterior, se definen las funciones de autocorrelación y autocovarianza para series de tiempo estacionarias de la siguiente manera:

Definición 2.3.3. La ***función autocovarianza*** (ACVF) de una serie de tiempo estacionaria $\{X_t : t \in T\}$, esta definida por:

$$\gamma_X(h) = \gamma_X(h,0) = \mathrm{Cov}(X_{t+h}, X_t)\ , \quad \text{para todo } \ t \in T.$$

Por otra parte, la ***función autocorrelación*** (ACF) se define como:

$$\rho_X(h) = \frac{\gamma_X(h)}{\gamma_X(0)} = \mathrm{Corr}(X_{t+h}, X_t)\ , \quad \text{para todo } \ t \in T.$$

Se enuncian algunas propiedades básicas de las funciones de autocovarianza y de autocorrelación de una serie de tiempo estacionaria.

Proposición 2.3.1. *Sea $\{X_t : t \in T\}$ una serie de tiempo estacionaria y $\gamma_X(h)$ la función autocovarianza de la serie, entonces $\gamma_X(h)$ cumple las siguientes propiedades:*

1. $\gamma_X(0) \geq 0$.
2. $|\gamma_X(h)| \leq \gamma_X(0)$, *para todo* h.
3. *La función $\gamma_X(h)$ es una función par.*

Demostración. La primera propiedad se tiene ya que $\gamma_X(0) = \mathrm{Cov}(X_t, X_t) = \mathrm{Var}(X_t)$ el cual es mayor o igual a cero. La segunda propiedad se obtiene de la desigualdad de Cauchy-Schwartz, la cual esta dada por: $(\mathrm{E}[XY])^2 \leq \mathrm{E}[X^2]\,\mathrm{E}[Y^2]$, así tomando $X = X_{t-h} - \mu$ y $Y = X_t - \mu$, se tiene que

$$\begin{aligned}\Big(\gamma_X(h)\Big)^2 &= \Big(\mathrm{E}[(X_{t+h} - \mu)(X_t - \mu)]\Big)^2 \\ &\leq \mathrm{E}[(X_{t+h} - \mu)(X_{t+h} - \mu)]\,\mathrm{E}[(X_t - \mu)(X_t - \mu)] \\ &= \Big(\gamma_X(0)\Big)^2\end{aligned}$$

Para la última propiedad se verifica al considerar que

$$\gamma_X(h) = \mathrm{Cov}(X_{t+h}, X_t) = \mathrm{Cov}(X_t, X_{t+h}) = \gamma_X(-h).$$

□

Proposición 2.3.2. *Sea $\{X_t : t \in T\}$ una serie de tiempo estacionaria, entonces la función de autocorrelación cumple lo siguiente:*

1. *La función de autocorrelación es una función par.*
2. *$|\rho_X(\tau)| \leq 1$.*
3. *La función de autocorrelación no es única.*

Demostración. Estas propiedades son consecuencias inmediatas de las propiedades de la función de autocovarianza, sólo recordando que la función de autocorrelación se define con respecto a ésta. □

En la practica $\gamma_X(h)$ y $\rho_X(h)$ son desconocidas y se deben estimar estos valores a partir de los datos que se dispongan $\{x_1, x_2, \ldots, x_k\}$. Además, para evaluar el grado de dependencia en los datos y poder seleccionar un modelo se requiere de una herramienta que ayude a realizarlo y para esto se utiliza la función de autocorrelación muestral.

Definición 2.3.4. Sea $\{x_t\}_{t=1}^n$ observaciones de una serie de tiempo. La función de ***autocorrelación muestral*** se define como

$$\hat{\gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x}),$$

con $\hat{\gamma}(-h) = \hat{\gamma}(h)$ para $h = 0, 1, \ldots, n-1$ y $\bar{x} = \frac{1}{n}\sum_{t=1}^n x_t$ el cual corresponde a la ***media muestral*** de $\{x_t\}_{t=1}^n$. La función de ***autocorrelación muestral*** está definida como

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}, \quad h < n.$$

Si se tiene la creencia de que los datos provienen de valores obtenidos de una serie de tiempo estacionaria $\{X_t : t \in T\}$, entonces la función de autocorrelación muestral brindará una estimación de la función de autocorrelación de $\{X_t : t \in T\}$. Esta estimación deberá de sugerir cual será el posible modelo de la serie para poder representar la dependencia en los datos.

2.4. Eliminación de las componentes

Como primer paso para realizar el análisis de una serie de tiempo es graficar el conjunto de datos. Si existen observaciones aisladas, estas deben de estudiarse de manera cuidadosa por la posibilidad de descartarlas (por ejemplo, si alguno de los datos fue obtenida de manera incorrecta). Al realizar una inspección del gráfico, se tiene la posibilidad de llevar a cabo una representación de los datos que involucren a los distintos tipos de variación de la serie.

Las series de tiempo *no estacionarias* suelen presentarse en procesos que se encuentran comúnmente en el mundo real, particularmente en la industria y economía en donde se esta interesado en la predicción de eventos o hechos. Para esto, el objetivo principal es la estacionalización de la serie para obtener modelos que permitan describir la serie dada, esto mediante la estimación y extracción de componentes deterministas, tales como m_t y s_t del modelo. Por ejemplo, la descomposición clásica convierte la componente ruido o fluctuación $\{\varepsilon_t\}$ en un proceso estocástico estacionario, luego se busca un modelo de probabilidad para $\{\varepsilon_t\}$ que permita describir y predecir, en conjunto de las componentes m_t y s_t, el proceso $\{X_t : t \in T\}$.

2.4.1. Eliminación mediante diferenciación

Definición 2.4.1. Se define el ***operador de retraso*** mediante

$$BX_t = X_{t-1} \ ,$$

el cual se extiende de manera natural en potencias; por ejemplo,

$$\begin{aligned} B^2X_t &= B(BX_t) \\ &= BX_{t-1} \\ &= X_{t-2} \ , \end{aligned}$$

así sucesivamente. Entonces, para cualquier potencia k se tiene

$$B^kX_t = X_{t-k} \ .$$

Definición 2.4.2. El ***operador de diferencia de orden*** d se define mediante

$$\nabla^d = (1 - B)^d,$$

donde el operador se puede expander de manera algebraica para poder evaluar la potencia d.

El objetivo de efectuar diferencias a la serie de tiempo es volverla estacionaria, sin embargo, al tratar una serie de tiempo que ya es estacionaria, esta seguirá siendo estacionaria, lo cual significaría que la serie se esta sobrediferenciando y esto puede generar problema como el de no poder identificar el modelo que pueda representarla, el incremento de la varianza (Véase [7]) y además perder observaciones al tratar con este tipo de operadores, ya que al aplicar dicho operador d veces, es decir ∇^d, se perderán automáticamente d observaciones de la serie.

Si se tiene un modelo de tendencia, $X_t = m_t + \varepsilon_t$, donde $m_t = \alpha + \beta t$, al aplicar el operador ∇ a la función m_t, entonces se tendrá una función constante $\nabla m_t = \beta$. Si la tendencia m_t es una función polinomial de grado k entonces es posible reducirlo a una función constante aplicando sucesivamente este operador k veces, es decir ∇^k. Por ejemplo,

si se supone que $m_t = \sum_{j=0}^{k} c_j t^j$ y ε_t es estacionario con media cero, al aplicar ∇^k se obtendrá

$$\nabla^k X_t = k! c_k + \nabla^k \varepsilon_t \ ,$$

donde $\nabla^k X_t$ es un proceso estacionario con media $k! c_k$.

Con lo anterior, es posible considerar una colección de datos $\{x_t\}$ que al aplicar el operador ∇ repetidamente se obtendrá una nueva colección $\{\nabla^k x_t\}$, el cual será una realización de un proceso estacionario. Es común en la práctica que el orden k del operador de diferencia sea pequeño, por ejemplo, $k = 1$ o $k = 2$.

2.5. Modelos estocásticos usados en series de tiempo

2.5.1. Modelo de media móvil

Definición 2.5.1. La serie de tiempo $\{X_t : t \in T\}$ es un proceso de ***media móvil de orden*** q, denotado por MA(q), si

$$X_t = \beta_0 Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q}, \tag{2.1}$$

donde $\{Z_t\} \sim \text{WN}(0, \sigma^2)$ y $\{\beta_i\}$ son constantes.

Este modelo, es ocupado en muchas áreas particularmente en econometría, donde los indicadores económicos son afectados por ciertos eventos aleatorios como son huelgas, decisiones políticas, entre otros. Aunque estos factores no afecten de manera inmediata a estos indicadores, el modelo MA(q) puede ser el apropiado para el ajuste de los datos. Además, otro punto importante en el uso del modelo MA(q) es que cualquier proceso que sea q-correlacionado es siempre un modelo MA(q), es decir, toda serie de tiempo X_t estacionaria con media cero que cumpla que $\gamma_X(q) = 0$ para $|h| > q$ es un modelo MA(q) (Véase [3]).

2.5.2. Modelos autorregresivos

Otro tipo de modelo que es usado habitualmente son los llamados *modelos autorregresivos.*

Definición 2.5.2. Una serie de tiempo $\{X_t : t \in T\}$ es un proceso ***autorregresivo de orden*** p, denotado por AR(p), si la serie puede ser escrita de la siguiente manera:

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + Z_t, \tag{2.2}$$

donde $\{Z_t\} \sim \text{WN}(0, \sigma^2)$ y $\{\alpha_i\}$ constantes.

Este modelo se asemeja al modelo de regresión múltiple, sin embargo la diferencia radica en que la variable X_t no es una regresión de las variables independientes, más bien su regresión depende totalmente de los valores que haya obtenido en el pasado.

2.5.3. Proceso Lineal

Las clases de modelos lineales de series de tiempo dan un panorama general en el estudio de los procesos estacionarios. Todo proceso estacionario de segundo orden es un proceso lineal o puede ser transformado a un proceso lineal, esto se realiza sustrayendo componentes *deterministas* al proceso, a este procedimiento se le da el nombre de *descomposición de Wold* (Véase [3]).

Definición 2.5.3. La serie de tiempo $\{X_t : t \in T\}$ es un ***proceso lineal***, si ésta tiene la siguiente representación:

$$X_t = \sum_{j=-\infty}^{\infty} \psi_j Z_{t-j}, \tag{2.3}$$

para todo t, donde $\{Z_t\} \sim \mathrm{WN}(0, \sigma^2)$ y además $\{\psi_j\}$ es una sucesion de constantes el cual cumple que $\sum_{j=-\infty}^{\infty} |\psi_j| < \infty$.

En términos del operador de retraso B, (2.3) puede ser reescrita como

$$X_t = \psi(B) Z_t, \tag{2.4}$$

donde $\psi(B) = \sum_{j=-\infty}^{\infty} \psi_j B^j$. Un proceso lineal es llamado *media móvil* o $\mathrm{MA}(\infty)$, si $\psi_j = 0$ para todo $j < 0$, es decir

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}.$$

El operador $\psi(B)$ de (2.4) puede ser visto como un *filtro lineal*, el cual al ser aplicado a la serie $\{Z_t\}$, produce como salida la serie $\{X_t\}$. El siguiente resultado asegura que al aplicar un filtro lineal a una serie que es estacionaria, la serie resultante sigue siendo estacionaria.

Proposición 2.5.1. *Sea $\{Y_t : t \in T\}$ una serie de tiempo estacionaria con media 0 y función de covarianza γ_Y. Si $\sum_{j=\infty}^{\infty} |\psi_j| < \infty$, entonces la serie de tiempo*

$$X_t = \sum_{j=-\infty}^{\infty} \psi_j Y_{t-j} = \psi(B) Y_t,$$

es estacionaria con media 0 y función de autocovarianza

$$\gamma_X(h) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \psi_j \psi_k \gamma_Y(h + k - j). \tag{2.5}$$

En el caso especial donde $\{X_t : t \in T\}$ sea un proceso lineal, se tiene que

$$\gamma_X(h) = \sum_{j=-\infty}^{\infty} \psi_j \psi_{j+h} \sigma^2. \tag{2.6}$$

DEMOSTRACIÓN. Como se tiene que $\mathrm{E}[Y_t] = 0$ para $t \in T$, entonces

$$\mathrm{E}[X_t] = E\left[\sum_{j=-\infty}^{\infty} \psi_j Y_{t-j}\right] = \left[\sum_{j=-\infty}^{\infty} \psi_j \,\mathrm{E}[Y_{t-j}]\right] = 0,$$

además

$$\begin{aligned}
\gamma_X(h) &= \mathrm{E}\left[(X_{t+h} - \mu)(X_t - \mu)\right] \\
&= \mathrm{E}[X_{t+h} X_t] \\
&= \mathrm{E}\left[\left(\sum_{j=-\infty}^{\infty} \psi_j Y_{t+h-j}\right)\left(\sum_{k=-\infty}^{\infty} \psi_k Y_{t-k}\right)\right] \\
&= \sum_{j=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} \psi_j \psi_k \,\mathrm{E}[Y_{t+h-j} Y_{t-k}] \\
&= \sum_{j=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} \psi_j \psi_k \gamma_Y(h - j + k).
\end{aligned}$$

De esto último, se tiene que $\{X_t : t \in T\}$ es estacionario y con función de autocovarianza (2.5). Ahora, si $\{X_t : t \in T\}$ es un proceso lineal, se tiene que $\{Y_t : t \in T\}$ es una sucesión de ruido blanco, así $\gamma_Y(h - j + k) = \sigma^2$ siempre que $k = j - h$ y 0 para cualquier otro caso, esto por la no correlación de la serie $\{Y_t : t \in T\}$. Así se concluye (2.6). □

2.5.4. Procesos autorregresivos de promedios móviles

Definición 2.5.4. Sea $\{X_t : t \in T\}$ una serie de tiempo. Se dice que es un ***proceso autorregresivo de promedios móviles***, denotado por ARMA(p, q), si $\{X_t : t \in T\}$ es estacionario y para todo t se tiene que:

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q}, \tag{2.7}$$

donde $\{Z_t\} \sim \mathrm{WN}(0, \sigma^2)$ y los polinomios $(1 - \phi_1 z - \cdots - \phi_p z^p)$ y $(1 + \theta_1 z + \cdots + \theta_q z^q)$ no tienen factores comunes.

El proceso $\{X_t : t \in T\}$ es un proceso ARMA(p, q) con media μ si $\{X_t - \mu : t \in T\}$ es un modelo ARMA(p, q). Una forma abreviada de expresar (2.7) es de la siguiente manera:

$$\phi(B) X_t = \theta(B) Z_t, \tag{2.8}$$

donde ϕ y θ son polinomios de orden p y q respectivamente, es decir,

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \qquad \text{y} \qquad \theta(z) = 1 - \theta_1 z - \cdots - \theta_q z^q,$$

además B es el operador de retraso. Como caso particular se tiene que la serie $\{X_t : t \in T\}$ es un proceso AR(p) si $\theta(z) \equiv 1$ y es un MA(q) si $\phi(z) \equiv 1$.

Una condición para la existencia de una solución para el proceso estacionario es que el polinomio $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$ sea distinto de cero para todo número complejo z con $|z| = 1$. Si $\phi(z) \neq 0$ para todo z en el circulo unitario, entonces existe $\delta > 0$ que cumple,

$$\frac{1}{\phi(z)} = \sum_{j=-\infty}^{\infty} \chi_j z^j, \quad \text{para } 1 - \delta < |z| < 1 + \delta,$$

y $\sum_{j=-\infty}^{\infty} |\chi_j| < \infty$. Definiendo entonces la parte derecha de la ecuación anterior como un filtro lineal cuyos coeficientes de los sumandos son positivos,

$$\frac{1}{\phi(B)} = \sum_{j=-\infty}^{\infty} \chi_j B^j,$$

y aplicando el operador $\chi(B) := \dfrac{1}{\phi(B)}$ en ambos lado de (2.8) se tiene entonces:

$$X_t = \chi(B)\phi(B)X_t = \chi(B)\theta(B)Z_t = \psi(B)Z_t = \sum_{j=-\infty}^{\infty} \psi_j Z_{t-j},$$

donde $\psi(z) = \chi(z)\theta(z) = \sum_{j=-\infty}^{\infty} \psi_j z^j$.

Proposición 2.5.2 (Existencia y unicidad)**.** *Una solución de (2.7) existe y además será única para la serie $\{X_t : t \in T\}$ si y sólo si,*

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \neq 0, \quad \textit{para todo } |z| = 1.$$

Proposición 2.5.3 (Causal)**.** *Un proceso* ARMA(p, q)*, $\{X_t : t \in T\}$ es* **causal** *o una* **función causal de** *$\{Z_t\}$ si existen contantes $\{\psi_j\}$ tal que $\sum_{j=0}^{\infty} |\psi_j| < \infty$ y además,*

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}, \quad \textit{para todo } t \in T. \tag{2.9}$$

Una condición equivalente para que la serie sea causal es

$$1 - \phi_1 z - \cdots - \phi_p z^p \neq 0, \quad \textit{para todo } |z| \leq 1. \tag{2.10}$$

La sucesión $\{\psi_j\}$ en (2.9) se determina por la relación $\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \dfrac{\theta(z)}{\phi(z)}$, o equivalentemente,

$$(1 - \phi_1 z - \cdots - \phi_p z^p)(\psi_0 + \psi_1 z + \dots) = 1 + \theta_1 z + \cdots + \theta_q z^q.$$

Igualando los coeficientes correspondientes a z^j con $j = 0, 1, \dots$, se encuentra que

$$\begin{aligned} 1 &= \psi_0 \\ \theta_1 &= \psi_1 - \psi_0\phi_1 \\ \theta_2 &= \psi_2 - \psi_1\phi_1 - \psi_0\phi_2 \\ &\vdots \end{aligned}$$

o equivalentemente,

$$\psi_j - \sum_{k=1}^{p} \phi_k \psi_{j-k} = \theta_j, \quad \text{con } \ j = 0, 1 \ldots,$$

donde $\theta_0 := 1, \theta_j := 0$ para $j > q$, además $\psi_j := 0$ para $j < 0$.

Proposición 2.5.4 (Invertibilidad)**.** *Un proceso* ARMA(p, q) $\{X_t : t \in T\}$ *es* **invertible** *si existen constantes* $\{\pi_j\}$ *tal que* $\sum_{j=0}^{\infty} |\pi_j| < \infty$ *y*

$$Z_t = \sum_{j=0}^{\infty} \pi_j X_{t-j}, \quad \textit{para todo } \ t.$$

La condición de invertibilidad es equivalente a:

$$\theta(z) = 1 - \theta_1 z + \cdots + \theta_q z^q \neq 0 \quad \textit{para todo, } \ |z| \leq 1.$$

Para poder obtener los valores de la sucesión $\{\pi_j\}$ se realiza el mismo razonamiento que fue utilizado para hallar los coeficientes de $\{\psi_j\}$. Así los valores serán determinados mediante la ecuación,

$$\pi_j + \sum_{k=1}^{q} \theta_k \pi_{j-k} = -\phi_j, \quad \text{para } \ j = 0, 1, \ldots,$$

donde $\phi_0 := -1, \phi_j := 0$ para $j > p$, además $\pi_j := 0$ para $j < 0$.

2.5.5. Función de Autocorrelación parcial

Definición 2.5.5. La ***función de autocorrelación parcial*** (PARC) de un proceso ARMA(p, q) $\{X_t : t \in T\}$ esta dada por la función $\alpha(\cdot)$ la cual esta definida mediante las siguientes ecuaciones

$$\alpha(0) = 1 \qquad \text{y} \qquad \alpha(h) = \phi_{hh}, \ \text{ con } \ h \leq 1,$$

donde ϕ_{hh} es la última componente correspondiente de

$$\phi_h = \Gamma_h^{-1} \gamma_h,$$

donde $\Gamma_h = [\gamma(i-j)]_{i,j=1}^{h}$ y $\gamma_h = [\gamma(1), \gamma(2), \ldots, \gamma(h)]'$.

Para un conjunto de observaciones $\{x_1, x_2, \ldots, x_n\}$ con $x_i \neq x_j$ para algún i y j, la ***función de autocorrelación parcial muestral***, $\hat{\alpha}(h)$ esta definido mediante:

$$\hat{\alpha}(0) = 1 \qquad \text{y} \qquad \hat{\alpha}(h) = \hat{\phi}_{hh} \ \text{ con } \ h \leq 1,$$

donde $\hat{\phi}_{hh}$ es la última componente correspondiente de

$$\hat{\phi}_h = \hat{\Gamma}_h^{-1} \hat{\gamma}_h.$$

La función de autocorrelación parcial mide la relación que existe en las observaciones hechas en la serie de tiempo separadas por un retraso de h unidades de tiempo eliminando el efecto de las observaciones intermedias.

2.6. Modelado y predicción de series de tiempo

Se considera el problema de predecir valores X_{n+h}, para $h > 0$ de una serie de tiempo estacionaria con media μ conocida y función de autocovarianza γ, en termino de los valores $\{X_n, \ldots, X_1\}$ hasta el tiempo n. El objetivo es hallar una combinación lineal de $1, X_n, X_{n-1}, \ldots, X_1$ que prediga X_{n+h} con el mínimo error cuadrático medio. El mejor predictor lineal que depende de $1, X_n, X_{n-1}, \ldots, X_1$ será denotado mediante

$$P_n X_{n+h} = a_0 + a_1 X_n + \cdots a_n X_1,$$

el cual queda determinado mediante los coeficientes $a_0, a_1, \ldots, a_n$ al buscar valores que minimicen

$$S(a_0, \ldots, a_n) = \mathrm{E}\left[X_{n+h} - a_0 - a_1 X_n - \cdots - a_n X_1\right]^2.$$

Donde S es una función cuadrática de $a_0, a_1, \ldots, a_n$ el cual esta acotado inferiormente por cero. Así existe un valor pequeño de $(a_0, \ldots, a_n)$ que minimiza a S y además satisface que

$$\frac{\partial S(a_0, \ldots, a_n)}{\partial a_j} = 0, \qquad j = 0, \ldots, n. \tag{2.11}$$

Evaluando estas derivadas en (2.11) se obtiene el siguiente sistema de ecuaciones:

$$\begin{aligned} &\mathrm{E}\left[X_{n+h} - a_0 - \sum_{i=1}^{n} a_i X_{n+1-i}\right] = 0, \\ &\mathrm{E}\left[\left(X_{n+h} - a_0 - \sum_{i=1}^{n} a_i X_{n+1-i}\right) X_{n+1-j}\right] = 0, \qquad j = 1, \ldots, n. \end{aligned} \tag{2.12}$$

Resolviendo el sistema se obtiene los valores de a_i que minimizan el error cuadrático medio, los cuales pueden ser reescritos en forma vectorial de la siguiente manera:

$$a_0 = \mu \left(1 - \sum_{i=1}^{n} a_i \right) \tag{2.13}$$

y

$$\Gamma_n \mathbf{a}_n = \gamma_n(h), \tag{2.14}$$

donde

$$\mathbf{a}_n = (a_1, \ldots, a_n)', \qquad \Gamma_n = [\gamma(i-j)]_{i,j=1}^{n},$$

y

$$\gamma_n(h) = \Big(\gamma(h), \gamma(h+1), \ldots, \gamma(h+n-1)\Big)'.$$

Así,

$$P_n X_{n+h} = \mu + \sum_{i=1}^{n} a_i \left(X_{n+1-i} - \mu \right), \tag{2.15}$$

donde $\mathbf{a}_n$ satisface (2.14). De (2.15) el valor esperado del error de predicción $X_{n+h} - P_n X_{n+h}$ es cero, y el error cuadrático medio de la predicción esta dado por:

$$\begin{aligned} \mathrm{E}[X_{n+h} - P_n X_{n+h}]^2 &= \gamma(0) - 2\sum_{i=1}^{n} a_i \gamma(h+i-1) + \sum_{i=1}^{n}\sum_{j=1}^{n} a_i \gamma(i-j) a_j \\ &= \gamma(0) - \mathbf{a}_n' \gamma_n(h), \end{aligned} \tag{2.16}$$

esta última ecuación se sigue por (2.14).

Proposición 2.6.1. *Sea $\{X_t : t \in T\}$ una serie de tiempo estacionaria con media μ, sea $P_n X_{n+h}$ el mejor predictor lineal definido como (2.12). Entonces $P_n X_{n+h}$ satisface las siguientes propiedades:*

1. $P_n X_{n+h} = \mu + \sum_{i=1}^{n} a_i \left(X_{n+1-i} - \mu \right)$, *donde* $\mathbf{a}_n = (a_1, \ldots, a_n)'$ *satisface (2.14).*
2. $\mathrm{E}[X_{n+h} - P_n X_{n+h}]^2 = \gamma(0) - \mathbf{a}_n' \gamma_n(h)$, *donde* $\gamma_n(h)$ *está definido mediante* $\gamma_n(h) = (\gamma(h), \ldots, \gamma(h+n-1))'$.
3. $\mathrm{E}[X_{n+h} - P_n X_{n+h}] = 0$.
4. $\mathrm{E}[(X_{n+h} - P_n X_{n+h}) X_j] = 0$, *para* $j = 1, \ldots, n$.

Demostración. Para el primer y segundo inciso, se dan de manera natural en la construcción del polinomio P_nX_{n+h}. Por otro lado, para el inciso tres se tiene que

$$\begin{aligned}\mathrm{E}[X_{n+h} - P_nX_{n+h}] &= \mathrm{E}[X_{n+h} - \mu + \sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)] \\ &= \mathrm{E}[X_{n+h}] - \mu + \sum_{i=1}^{n} a_i\,\mathrm{E}[X_{n+1-i} - \mu] \\ &= 0.\end{aligned}$$

Para el último inciso, se tiene que:

$$\begin{aligned}\mathrm{E}[(X_{n+h} - P_nX_{n+h})X_j] &= \mathrm{E}\left[\left(X_{n+h} - \mu + \sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)\right)X_j\right] \\ &= \mathrm{E}\left[X_{n+h}X_j - \mu X_j - \sum_{i=1}^{n} a_i\,(X_{n+1-i} - \mu)\,X_j\right] \\ &= E\,[X_{n+h}X_j] - \mu^2 - \sum_{i=1}^{n} a_i\left(\mathrm{E}[X_{n+1-i}X_j] - \mu^2\right) \\ &= \mathrm{E}\Big[(X_{n+h} - \mu)(X_j - \mu)\Big] - \sum_{i=1}^{n} a_i\Big(\mathrm{E}[(X_{n+1-i} - \mu)(X_j - \mu)]\Big) \\ &= \gamma(n + h - j) - \sum_{i=1}^{n} a_i\gamma(n + 1 - j - i).\end{aligned}$$

Observe que de (2.14) se tiene que para $j = 1, \ldots, n$ se cumple

$$\gamma(h - 1 + j) = \sum_{i=1}^{n} a_i\gamma(i - j),$$

como $\gamma(i - j) = \gamma(j - i)$, se tiene entonces que

$$\begin{aligned}\sum_{i=1}^{n} a_i\gamma\Big((n + 1 - j) - i\Big) &= \gamma\Big(h - 1 + (n + 1 - j)\Big) \\ &= \gamma(h + n - j).\end{aligned}$$

luego $\mathrm{E}[(X_{n+h} - P_nX_{n+h})X_j] = 0$ para $j = 1, 2, \ldots, n$.

□

2.6.1. Algoritmo *Innovations*

El algoritmo *innovations* es un algoritmo recursivo el cual es aplicable a cualquier tipo de serie de tiempo con segundo momento finito e independientemente de que la serie sea o no estacionaria. Supóngase que la serie $\{X_t : t \in T\}$ tiene media cero y además $\mathrm{E}\,|X_t|^2 < \infty$ para cada t y $\mathrm{E}[X_i X_j] = \kappa(i,j)$. Definase ahora el ***mejor predictor de un paso*** y ***el error cuadrático medio mediante***

$$\hat{X}_n = \begin{cases} 0, & \text{si } n = 1; \\ P_{n-1}X_n, & \text{si } n = 2, 3, \ldots, \end{cases}$$

y

$$v_n = \mathrm{E}[X_{n+1} - P_n X_{n+1}]^2,$$

donde $P_{n-1}X_n$ se define como el ***mejor predictor lineal*** en términos de $1, X_{n-1}, \ldots, X_1$.

Se define ahora el ***innovations***, o el *error de predicción de un paso* como

$$U_n = X_n - \hat{X}_n.$$

Con lo anterior, en termino de vectores se tendrá que $\mathbf{U}_n = (U_1, \ldots, U_n)'$ y $\mathbf{X}_n = (X_1, \ldots, X_n)'$. Luego, la última ecuación puede ser reescrita como

$$\mathbf{U}_n = A_n \mathbf{X}_n. \tag{2.17}$$

donde

$$A_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_{11} & 1 & 0 & \cdots & 0 \\ a_{22} & a_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ a_{n-1,n-1} & a_{n-1,n-2} & a_{n-1,n-3} & \cdots & 1 \end{bmatrix}.$$

La matriz A_n es no singular cuya inversa C_n es de la forma

$$C_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \theta_{11} & 1 & 0 & \cdots & 0 \\ \theta_{22} & \theta_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \theta_{n-1,n-1} & \theta_{n-1,n-2} & \theta_{n-1,n-3} & \cdots & 1 \end{bmatrix}. \tag{2.18}$$

Por tanto, el vector predictor de un paso $\hat{\mathbf{X}}_n := (X_1, P_1X_2, \ldots, P_{n-1}X_n)'$ se puede expresar como:

$$\begin{aligned}\hat{\mathbf{X}}_n =& \mathbf{X}_n - \mathbf{U}_n \\ =& C_n \mathbf{U}_n - \mathbf{U}_n \\ =& (C_n - I_n)\,\mathbf{U}_n \\ =& (C_n - I_n)\left(\mathbf{X}_n - \hat{\mathbf{X}}_n\right) \\ =& \boldsymbol{\Theta}_n \left(\mathbf{X}_n - \hat{\mathbf{X}}_n\right),\end{aligned} \tag{2.19}$$

donde,

$$\boldsymbol{\Theta}_n = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \theta_{11} & 0 & 0 & \cdots & 0 \\ \theta_{22} & \theta_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \theta_{n-1,n-1} & \theta_{n-1,n-2} & \theta_{n-1,n-3} & \cdots & 0 \end{bmatrix},$$

y $\mathbf{X}_n$ satisface también

$$\mathbf{X}_n = C_n(\mathbf{X}_n - \hat{\mathbf{X}}_n). \tag{2.20}$$

Así (2.17) se puede reescribir como:

$$\hat{X}_{n+1} = \begin{cases} 0, & \text{si } n = 0; \\ \displaystyle\sum_{j=1}^{n} \theta_{nj}(X_{n+1-j} - \hat{X}_{n+1-j}), & \text{si } n = 1, 2, \ldots. \end{cases} \tag{2.21}$$

Los predictores de un paso $\hat{X}_1, \hat{X}_2, \ldots$ se puede calcular recursivamente una vez que se hayan obtenido los coeficientes θ_{ij} los cuales se calculan de la siguiente manera.

Definición 2.6.1. Los coeficientes $\theta_{n1}, \theta_{n2}, \ldots, \theta_{nn}$ se calculan de manera recursiva por medio de las siguientes ecuaciones:

$$v_0 = \kappa(1,1),$$

$$\theta_{n,n-k} = v_k^{-1}\left(\kappa(n+1,k+1) - \sum_{j=0}^{k-1} \theta_{k,k-j}\theta_{n,n-j}v_j\right), \quad 0 \le k \le n,$$

$$v_n = \kappa(n+1,n+1) - \sum_{j=0}^{n-1} \theta_{n,n-j}^2 v_j.$$

Cálculo recursivo de predictores en h-pasos

Para la predicción en h-pasos se utiliza la siguiente igualdad (Véase [3])

$$P_n(X_{n+k} - P_{n+k-1}X_{n+k}) = 0, \qquad k \geq 1. \tag{2.22}$$

lo cual se sigue de

$$\mathrm{E}[(X_{n+k} - P_{n+k-1}X_{n+k} - 0)\, X_{n+j-1}] = 0, \qquad j = 1, \ldots, n. \tag{2.23}$$

Utilizando la linealidad de P_n en (2.22) se tiene que

$$\begin{aligned} P_n X_{n+h} &= P_n P_{n+h-1} X_{n+h} \\ &= P_n \hat{X}_{n+h} \\ &= P_n \left(\sum_{j=1}^{n+h-1} \theta_{n+h-1,j} \left(X_{n+h-1} - \hat{X}_{n+h-j} \right) \right) \\ &= \sum_{j=1}^{n+h-1} \theta_{n+h-1,j}\, P_n(X_{n+h-j} - P_{n+h-j-1}X_{n+h-j}), \end{aligned}$$

para $h - j \geq 1$, $P_n(X_{n+h-j} - P_{n+h-j-1}X_{n+h-j}) = 0$ (esto por (2.22)), así

$$P_n X_{n+h} = \sum_{j=h}^{n+h-1} \theta_{n+h-1,j} \left(X_{n+h-j} - \hat{X}_{n+h-j} \right), \tag{2.24}$$

donde los coeficientes θ_{nj} son determinados por el algoritmo *innovations*.

El error medio cuadrático esta dado mediante

$$\mathrm{E}[X_{n+h} - P_n X_{n+h}]^2 = \mathrm{E}[X_{n+h}]^2 - 2\,\mathrm{E}[X_{n+h} P_n X_{n+h}] + \mathrm{E}[(P_n X_{n+h})^2],$$

usando el hecho de que $\mathrm{E}[X_{n+h} P_n X_{n+h}] = \mathrm{E}[(P_n X_{n+h})^2]$, entonces se tiene que

$$\mathrm{E}[X_{n+h} - P_n X_{n+h}]^2 = \mathrm{E}[X_{n+h}]^2 - \mathrm{E}[P_n X_{n+h}]^2,$$

se puede mostrar que $X_n - \hat{X}_n$ es no correlacionado respecto a $X_1 - \hat{X}_1, \ldots, X_{n-1} - \hat{X}_{n-1}$ (Véase [3]), así utilizando esto y usando la definición de v_k se concluye que

$$\mathrm{E}[X_{n+h} - P_n X_{n+h}]^2 = \kappa(n+h, n+h) - \sum_{j=h}^{n+h-1} \theta^2_{n+h-1,j} v_{n+h-j-1}. \tag{2.25}$$

2.6.2. Prediccion en procesos ARMA

Para poder determinar un modelo apropiado ARMA(p, q) que pueda describir una serie de tiempo observada involucra un conjunto de problemas relacionados entre sí. Esto incluye la elección de p y q (selección de orden) además de la estimación de la media, los coeficientes $\{\phi_i : i = 1, 2 \ldots, p\}$, $\{\theta_i : i = 1, 2 \ldots, q\}$ incluyendo la varianza del ruido blanco σ^2.

El algoritmo *innovations* provee un algoritmo recursivo para el pronóstico de un proceso con media cero el cual no necesariamente sea estacionario.

Para un proceso un proceso ARMA(p, q) causal $\{X_t : t \in T\}$ con media cero definido como

$$\phi(B)X_t = \theta(B)Z_t, \quad \{Z_t\} \sim \mathrm{WN}(0, \sigma^2). \tag{2.26}$$

Puede ser aplicado el algoritmo anteriormente mencionado usando el proceso $\{W_t\}$, el cual esta definido por:

$$W_t = \begin{cases} \sigma^{-1}X_t, & t = 1, 2, \ldots, m; \\ \\ \sigma^{-1}\phi(B)X_t, & t > m. \end{cases} \tag{2.27}$$

donde

$$m = \max(p, q). \tag{2.28}$$

Para este caso, se define $\theta_0 = 1$ y $\theta_j = 0$ para $j > q$, además se hace el supuesto de que $p \geq 1$ y $q \geq 1$. La autocovarianza $\kappa(i, j) = \mathrm{E}[W_i W_j]$, $i, j \geq 1$, esta puede ser calculada mediante la siguiente proposición.

Proposición 2.6.2. *Supongase que se tiene $\{X_t : t \in T\}$ un proceso* ARMA(p, q) *causal con $p \geq 1$ y $q \geq 1$ con media cero. Sea W_t definido como (2.27) además $\theta_0 = 1$ y $\theta_j = 0$ para $j > q$ entonces la función de autocovarianza esta dada mediante:*

$$\kappa(i, j) = \begin{cases} \sigma^{-2}\gamma_X(i - j), & 1 \leq i, j \leq m; \\ \\ \sigma^{-2}\left[\gamma_X(i - j) - \displaystyle\sum_{r=1}^{p} \phi_r \gamma_X(r - |i - j|)\right], & \min(i, j) \leq m < \max(i, j) \leq 2m; \\ \\ \displaystyle\sum_{r=0}^{q} \theta_r \theta_{r+|i-j|}, & \min(i, j) > m; \\ \\ 0, & \textit{otro caso.} \end{cases} \tag{2.29}$$

Demostración.

CASO I

Sea $1 \leq i, j \leq m$, luego

$$\begin{aligned}\kappa(i,j) &= \mathrm{E}\left[W_i W_j\right] \\ &= \mathrm{E}\left[\sigma^{-1} X_i \sigma^{-1} X_j\right] \\ &= \sigma^{-2}\,\mathrm{E}\left[X_i X_j\right] \\ &= \sigma^{-2}\gamma_X\left(i-j\right).\end{aligned}$$

CASO II

Suponga que mín$(i,j) = i$ luego máx$(i,j) = j$ entonces se sigue que:

$$\begin{aligned}\mathrm{E}(W_i W_j) &= \mathrm{E}\left[\sigma^{-1} X_i \sigma^{-1} \phi(B) X_j\right] \\ &= \sigma^{-2}\,\mathrm{E}\left[X_i \phi(B) X_j\right] \\ &= \sigma^{-2}\,\mathrm{E}\left[X_i\left(X_j - \phi_1 X_{j-1} - \phi_2 X_{j-2} - \cdots - \phi_p X_{j-p}\right)\right] \\ &= \sigma^{-2}\left(\mathrm{E}\left[X_i X_j - \sum_{r=1}^{p} \phi_r X_i X_{j-r}\right]\right) \\ &= \sigma^{-2}\left(\mathrm{E}\left[X_i X_j\right] - \sum_{r=1}^{p} \phi_r\,\mathrm{E}\left[X_i X_{j-r}\right]\right) \\ &= \sigma^{-2}\left(\gamma_X(i-j) - \sum_{r=1}^{p} \phi_r \gamma_X(j-k-i)\right),\end{aligned}$$

puesto que $\gamma_X(-h) = \gamma_X(h)$, entonces la última ecuación se expresa como:

$$\kappa(i,j) = \sigma^{-2}\left(\gamma_X(i-j) - \sum_{r=1}^{p} \phi_r \gamma_X(k-(j-i))\right).$$

CASO III

Supongase que mín$(i,j) = i$, luego $i > m$ y se tiene

$$\begin{aligned}\mathrm{E}[W_i W_j] &= \mathrm{E}\left[\sigma^{-1}\phi(B) X_i \sigma^{-1}\phi(B) X_j\right] \\ &= \sigma^{-2}\,\mathrm{E}\left[\phi(B) X_i \phi(B) X_j\right] \\ &= \sigma^{-2}\,\mathrm{E}\left[\theta(B) Z_i \theta(B) Z_j\right] \\ &= \sigma^{-2}\,\mathrm{E}\left[\left(Z_i + \theta_1 Z_{i-1} + \cdots + \theta_q Z_{i-q}\right)\left(Z_j + \theta_1 Z_{j-1} + \cdots + \theta_q Z_{j-q}\right)\right] \\ &= \sigma^{-2}\,\mathrm{E}\left[Z_i Z_j + \sum_{k=1}^{q} \theta_k Z_i Z_{j-k} + \cdots + \sum_{k=1}^{q} \theta_q \theta_k Z_{i-q} Z_{j-k}\right],\end{aligned}$$

puesto que $\{Z_t\} \sim \mathrm{WN}(0, \sigma^2)$, esto significa que $\{Z_t\}$ es no correlacionado es decir,

$$\mathrm{E}[Z_i Z_j] = \begin{cases} 0, & \text{si } i \neq j; \\ \sigma^2, & \text{si } i = j. \end{cases} \tag{2.30}$$

Bajo el supuesto de que $\min(i, j) = i$ entonces se tiene que $i < j$, luego existe $h > 0$ tal que $i + h = j$ y usando (2.30) se tiene entonces

$$\begin{aligned} \mathrm{E}[W_i W_j] &= \sigma^{-2}\Big(\theta_h \,\mathrm{E}[Z_i Z_{j-h}] - \theta_1\theta_{h+1}\,\mathrm{E}[Z_{i-1} Z_{j-h+1}] - \cdots - \theta_q\theta_{h+q}\,\mathrm{E}[Z_{i-q} Z_{j-h+q}]\Big) \\ &= \theta_h - \theta_1\theta_{h+1} - \cdots - \theta_q\theta_{h+q} \\ &= \sum_{r=0}^{q} \theta_r \theta_{r+(j-i)}, \end{aligned}$$

donde $\theta_0 = 1$. Análogamente, se realiza el mismo procedimiento para el caso cuando $\min(i, j) = j$.

CASO IV

Para este caso, se tiene que $\min(i, j) > m$ y además $|i-j| > q$ entonces del caso anterior, existe $h > 0$ tal que $h = q + k$ para todo $k > 0$, sin embargo por hipótesis se tiene que $\theta_i = 0$ siempre que $i > q$, así se concluye que

$$\mathrm{E}[W_i W_j] = 0$$

□

Al aplicar el algoritmo *innovations* al proceso $\{W_t\}$, se tiene que

$$\hat{W}_{n+1} = \begin{cases} \displaystyle\sum_{j=1}^{n} \theta_{nj} \left(W_{n+1-j} - \hat{W}_{n+1-j}\right), & 1 \leq n < m; \\ \displaystyle\sum_{j=1}^{q} \theta_{nj} \left(W_{n+1-j} - \hat{W}_{n+1-j}\right), & n \geq m, \end{cases} \tag{2.31}$$

donde los coeficientes θ_{nj} y el error cuadrático medio $r_n = \mathrm{E}[W_{n+1} - \hat{W}_{n+1}]^2$ se encuentran de forma recursiva por medio de dicho algoritmo.

Cabe destacar que la función predictora (2.31) se hace cero por medio de θ_{nj} donde $n \geq m$ y $j > q$ el cual es consecuencia de la forma en que se da la función $\kappa(i, j)$, (proposición 2.6.2).

Se puede observar que en (2.27) cada X_n con $n \geq 1$ puede ser escrito como una combinación lineal de W_j con $1 \leq j \leq n$ y recíprocamente cada W_n con $n \geq 1$ puede escribirse como una combinación lineal de X_j para $1 \leq j \leq n$, luego el mejor predictor lineal de cualquier

variable aleatoria Y en términos de $\{1, X_1, \ldots, X_n\}$ es el mismo que el mejor predictor lineal de la variable Y en términos de $\{1, W_1, \ldots, W_n\}$. Así se tendrá entonces que los mejores predictores lineales de W_{n+1} y X_{n+1} de un paso están dados por:

$$\hat{W}_{n+1} = P_n W_{n+1} \qquad \text{y} \qquad \hat{X}_{n+1} = P_n X_{n+1},$$

respectivamente.

Usando lo anterior y (2.27), se observa que:

$$\hat{W}_t = \begin{cases} \sigma^{-1}\hat{X}_t, & t = 1, 2, \ldots, m; \\ \sigma^{-1}\left[\hat{X}_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p}\right], & t > m, \end{cases} \tag{2.32}$$

usando esta ecuación junto con (2.27), se tiene;

$$X_t - \hat{X}_t = \sigma\left(W_t - \hat{W}_t\right) \quad \text{para todo } t \geq 1. \tag{2.33}$$

Realizando el reemplazo de $(W_j - \hat{W}_j)$ mediante $\sigma^{-1}(X_j - \hat{X}_j)$ en (2.31) y haciendo después la sustitución en (2.32) se obtiene:

$$\hat{X}_{n+1} = \begin{cases} \displaystyle\sum_{j=1}^{n} \theta_{nj}\left(X_{n+1-j} - \hat{X}_{n+1-j}\right), & 1 \leq n \leq m; \\ \phi_1 X_n + \cdots + \phi_p X_{n+1-p} + \displaystyle\sum_{j=1}^{q} \theta_{nj}\left(X_{n+1-j} - \hat{X}_{n+1-j}\right), & n \geq m. \end{cases} \tag{2.34}$$

y además,

$$\mathrm{E}\left[X_{n+1} - \hat{X}_{n+1}\right]^2 = \sigma^2 \,\mathrm{E}\left[W_{n+1} - \hat{W}_{n+1}\right]^2 = \sigma^2 r_n. \tag{2.35}$$

Con esto se determina la predicción de un paso $\hat{X}_2, \hat{X}_3, \ldots$ recursivamente para un modelo ARMA(p, q) causal.

Predicción de procesos ARMA(p, q) **en** h**-pasos**

De (2.24) se tiene que

$$\begin{aligned} P_n W_{n+h} &= \sum_{j=h}^{n+h-1} \theta_{n+h-1,j}\left(W_{n+h-j} - \hat{W}_{n+h-j}\right) \\ &= \sigma^{-1} \sum_{j=h}^{n+h-1} \theta_{n+h-1,j}\left(X_{n+h-j} - \hat{X}_{n+h-j}\right). \end{aligned}$$

Usando el resultado anterior y aplicando el operador P_n a cada lado de (2.27) se concluye que el predictor en h-pasos P_{n+h} satisface:

$$P_n X_{n+h} = \begin{cases} \displaystyle\sum_{j=h}^{n+h-1} \theta_{n+h-1,j} \left(X_{n+h-j} - \hat{X}_{n+h-j} \right), & 1 \leq h \leq m-n; \\ \displaystyle\sum_{i=1}^{p} \phi_i P_n X_{n+h-i} + \sum_{j=h}^{n+h-1} \theta_{n+h-1,j} \left(X_{n+h-j} - \hat{X}_{n+h-j} \right), & h > m-n. \end{cases} \tag{2.36}$$

Para el caso de que $n > m$, entonces para todo $h \geq 1$,

$$P_n X_{n+h} = \sum_{i=1}^{p} \phi_i P_n X_{n+h-i} + \sum_{j=h}^{q} \theta_{n+h-1,j} \left(X_{n+h-j} - \hat{X}_{n+h-j} \right). \tag{2.37}$$

Una vez que los predictores $\hat{X}_1, \ldots, \hat{X}_n$ han sido calculados de (2.34), es sencilla la obtención (con n fijo) de los predictores $P_n X_{n+1}, P_n X_{n+2}, P_n X_{n+3}, \ldots$.

El error medio cuadrático de $P_n X_{n+h}$ se calcula mediante (Véase [3])

$$\sigma_n^2(h) = \mathrm{E}[X_{n+h} - P_n X_{n+h}]^2 = \sum_{j=0}^{h-1} \left(\sum_{r=0}^{j} \chi_r \theta_{n+h-r-1,j-r} \right)^2 v_{n+h-j-1}$$

donde los coeficientes χ_j son calculados de manera recursiva, con $\chi_0 = 1$ y para $j = 1, 2, \ldots$

$$\chi_j = \sum_{k=1}^{\min(p,j)} \phi_k \chi_{j-k} \tag{2.38}$$

2.6.3. Estimación por máxima verosimilitud

Suponga que $\{X_t : t \in T\}$ es una serie de tiempo gaussiana[1] con media cero y función de autocovarianza $\kappa(i,j) = \mathrm{E}[X_i X_j]$. Sean $\mathbf{X}_n = (X_1, \ldots, X_n)'$ y $\hat{\mathbf{X}}_n = (\hat{X}_1, \ldots, \hat{X}_n)'$, donde $\hat{\mathbf{X}}_1 = 0$ y $\hat{\mathbf{X}}_j = \mathrm{E}[X_j | X_1, \ldots, X_{j-1}] = P_{j-1} X_j$ con $j \geq 2$. Sea Γ_n la matriz de covarianzas, esto es $\Gamma_n = \mathrm{E}[\mathbf{X}_n \mathbf{X}_n']$ y suponiendo que la matriz Γ_n es no singular se tiene que la verosimilitud de $\mathbf{X}_n$ esta dada mediante:

$$L(\Gamma_n) = (2\pi)^{-n/2} (\det \Gamma_n)^{-1/2} \exp\left(-\frac{1}{2} \mathbf{X}_n' \Gamma_n^{-1} \mathbf{X}_n \right). \tag{2.39}$$

El calculo directo del $\det \Gamma_n$ y Γ^{-1} puede evitarse expresando estos en términos de los errores de predicción de un paso $X_j - \hat{X}_j$ junto con sus varianzas v_{j-1} para $j = 1, \ldots, n$ los cuales se obtiene de manera recursiva mediante el algoritmo *innovations*.

[1] Esto es, una serie de tiempo cuya función de distribución conjunta es normal multivariada

Para esto, sea θ_{ij} con $j = 1, 2, \ldots; i = 1, 2, \ldots$ los coeficientes que son obtenidos cuando se aplica el algoritmo *innovations* a la función de autocovarianza κ a la serie $\{X_t : t \in T\}$ además sea C_n la matriz triangular inferior de $n \times n$ definida en (2.18). Entonces, se tendrá que:

$$\mathbf{X}_n = C_n \left(\mathbf{X}_n - \hat{\mathbf{X}}_n \right).$$

Luego,

$$\begin{aligned} \Gamma_n &= \mathrm{E}\left[\mathbf{X}_n \mathbf{X}_n'\right] \\ &= \mathrm{E}\left[C_n \left(\mathbf{X}_n - \hat{\mathbf{X}}_n \right) \left(\mathbf{X}_n - \hat{\mathbf{X}}_n \right)' C_n' \right] \\ &= C_n \, \mathrm{E}\left[\left(\mathbf{X}_n - \hat{\mathbf{X}}_n \right) \left(\mathbf{X}_n - \hat{\mathbf{X}}_n \right)' \right] C_n'. \end{aligned}$$

Puesto que las componentes de $\mathbf{X}_n - \hat{\mathbf{X}}_n$ son no correlacionadas se tiene que la matriz de covarianzas es diagonal cuyos elementos son los v_j, es decir

$$D_n = \mathrm{diag}\{v_0, v_1, \ldots, v_{n-1}\}.$$

De esto último se obtendrá que

$$\Gamma_n = C_n D_n C_n'.$$

Por otra parte, se tiene que

$$\begin{aligned} \mathbf{X}_n' \Gamma_n^{-1} \mathbf{X}_n &= \mathbf{X}_n' \left(C_n D_n C_n' \right)^{-1} \mathbf{X}_n \\ &= \mathbf{X}_n' C_n'^{-1} D_n^{-1} C_n^{-1} \mathbf{X}_n \\ &= \left(\mathbf{X}_n' A_n' \right) D_n^{-1} \left(A_n \mathbf{X}_n \right) \\ &= \left(A_n \mathbf{X}_n \right)' D_n^{-1} \left(A_n \mathbf{X}_n \right) \\ &= \left(\mathbf{X}_n - \hat{\mathbf{X}}_n \right)' D_n^{-1} \left(\mathbf{X}_n - \hat{\mathbf{X}}_n \right) \\ &= \sum_{j=1}^{n} \frac{1}{v_{j-1}} \left(X_j - \hat{X}_j \right)^2. \end{aligned} \tag{2.40}$$

Además,

$$\begin{aligned} \det \Gamma_n &= \det \left(C_n D_n C_n' \right) \\ &= \det C_n \det D_n \det C_n' \\ &= \det D_n \\ &= v_0 v_1 \cdots v_n. \end{aligned} \tag{2.41}$$

De lo anterior, se observa que (2.39) se reduce a

$$L(\Gamma_n) = \frac{1}{\sqrt{(2\pi)^n v_0 v_1 \cdots v_{n-1}}} \exp\left\{ -\frac{1}{2} \sum_{j=1}^{n} \frac{\left(X_j - \hat{X}_n\right)^2}{v_{j-1}} \right\}. \tag{2.42}$$

Teniendo en cuenta (2.34) y (2.35), la expresión (2.42) se puede reescribir como sigue

Verosimilitud Gaussiana para un proceso ARMA

$$L(\phi, \theta, \sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)^n r_0 r_1 \ldots r_{n-1}}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{j=1}^{n} \frac{\left(X_j - \hat{X}_j\right)^2}{r_{j-1}} \right\}. \tag{2.43}$$

Al diferenciar $\ln L(\phi, \theta, \sigma^2)$ parcialmente respecto a σ^2 y observando que $\hat{X}_j$ y r_j son independientes de σ^2, se hallan los estimadores de máxima verosimilitud $\hat{\phi}, \hat{\theta}$ y σ^2 que satisface las siguientes propiedades:

Estimadores de máxima verosimilitud

$$\hat{\sigma}^2 = n^{-1} S\left(\hat{\phi}, \hat{\theta}\right), \tag{2.44}$$

donde

$$S\left(\hat{\phi}, \hat{\theta}\right) = \sum_{j=1}^{n} \frac{\left(X_j - \hat{X}_j\right)}{r_{j-1}}, \tag{2.45}$$

y $\hat{\phi}, \hat{\theta}$ son los valores de ϕ, θ que minimizan

$$\ell(\phi, \theta) = \ln\left(n^{-1} S(\phi, \theta)\right) + n^{-1} \sum_{j=1}^{n} \ln r_{j-1}. \tag{2.46}$$

Los estimadores de mínimos cuadrados $\tilde{\phi}$ y $\tilde{\theta}$ de ϕ y θ respectivamente, son obtenidos al minimizar la función S la cual se define mediante (2.45) en lugar de ℓ, la cual esta definida en (2.46) y sujeto a las restricciones de que el modelo sea causal e invertible. El estimador por mínimos cuadrados de σ^2 esta definido por

$$\tilde{\sigma}^2 = \frac{S\left(\tilde{\phi}, \tilde{\theta}\right)}{n - p - q}.$$

2.7. Diagnóstico y verificación

Usualmente, la bondad del ajuste de un modelo estadístico en un conjunto de datos es juzgado al comparar los valores observados con los correspondientes valores predichos los cuales fueron obtenidos del modelo ajustado.

Cuando se ajusta un modelo ARMA(p,q) a una serie dada éste determinará estimadores $\hat{\phi}, \hat{\theta}$ y $\hat{\sigma}^2$ de máxima verosimilitud respecto a los parámetros ϕ, θ y σ^2 respectivamente.

Se define los ***residuos*** mediante

$$\hat{W}_t = \frac{X_t - \hat{X}_t\left(\hat{\phi}, \hat{\theta}\right)}{\left(r_{t-1}\left(\hat{\phi}, \hat{\theta}\right)\right)^{1/2}}, \quad t = 1, \ldots, n\ , \tag{2.47}$$

donde $\hat{X}_t(\hat{\theta}, \hat{\phi})$ denota a los valores predichos de X_t basado en $X_1, X_2, \cdots, X_{t-1}$ los cuales son calculados al ajustar el modelo.

Si se asume que el modelo de máxima verosimilitud ARMA(p,q) genera a $\{X_t : t \in T\}$, entonces se tiene que $\{\hat{W}_t\} \sim \text{WN}(0, \sigma^2)$. Sin embargo, para mostrar que el modelo ARMA(p,q) sea el apropiado para los datos se asume que sólo $X_1, \ldots, X_n$ son generados y cuyos estimadores de máxima verosimilitud son $\hat{\phi}$, $\hat{\theta}$ y $\hat{\sigma^2}$, respectivamente por un proceso ARMA(p,q) cuyos parámetros ϕ, θ y σ^2 son desconocidos. Entonces $\{W_t\}$ no tiene una distribución de ruido blanco. Sin embargo $\hat{W}_t$ para $t = 1, \ldots, n$ tendrá propiedades similares a la secuencia de ruido blanco

$$W_t(\phi, \theta) = \frac{X_t - \hat{X}_t(\phi, \theta)}{(r_{t-1}(\phi, \theta))^{1/2}}, \quad t = 1, \ldots, n. \tag{2.48}$$

Por otra parte $W_t(\phi, \theta)$ se aproxima en termino al ruido blanco en la definición de (2.26) en el sentido de que $\text{E}[W_t(\phi, \theta) - Z_t]^2 \to 0$ cuando $t \to \infty$ (Véase [4]). Así la sucesión $\{\hat{W}_t\}$ deberá ser, aproximadamente

- No correlacionado si $\{Z_t\} \sim \text{WN}(0, \sigma^2)$.
- Independiente si $\{Z_t\} \sim \text{IID}(0, \sigma^2)$.
- Distribuida normalmente si $\{Z_t\} \sim \text{N}(0, \sigma^2)$.

El ***residuo reescalado*** $\hat{R}_t, t = 1, \ldots, n$ se obtiene al dividir los residuos $\hat{W}_t,\ t = 1, \ldots, n$ por el estimador $\hat{\sigma} = \sqrt{\dfrac{\sum_{t=1}^{n} W_t^2}{n}}$ del ruido blanco. Esto es

$$\hat{R}_t = \frac{\hat{W}_t}{\hat{\sigma}}. \tag{2.49}$$

Si el modelo es el apropiado, los residuos escalares deberán tener propiedades similares a las de una secuencia WN$(0,1)$ o de una secuencia IID$(0,1)$ si se hace el supuesto de que el ruido blanco $\{Z_t\}$ derivado del proceso ARMA es independiente del ruido blanco.

Lo siguiente está basado en las propiedades esperadas de los residuos o residuos reescalares bajo el supuesto de que el modelo ajustado es el correcto y además $\{Z_t\} \sim \text{IID}(0,\sigma^2)$.

Gráfica de los residuales reescalados $\{\hat{R}_t, t = 1,\ldots,n\}$

Si el modelo ajustado es el apropiado, entonces la gráfica de los residuos reescalados $\{\hat{R}_t, t = 1,\ldots,n\}$ deberá asemejarse a una secuencia de ruido blanco con varianza uno. Si bien es difícil el identificar la estructura de la correlación de $\{\hat{R}_t\}$ para esta gráfica, puesto que la desviación de la media en cero es a veces indicado por la componente de tendencia o componente cíclica y la varianza, la cual es no constante debido por la fluctuaciones en $\hat{R}_t$ cuya magnitud depende de t.

La función de autocorrelación muestral de los residuales

Para n lo suficientemente grande se tiene que la función de autocorrelación muestral de una sucesión i. i. d., $Y_1,\ldots,Y_n$ con varianza finita es aproximadamente i. i. d. cuya distribución es N$(0,1/n)$. De esto se puede realizar una prueba de hipótesis sobre los residuales observados los cuales son consistentes con un ruido i. i. d., esto al examinar las autocorrelaciones muestrales de los residuos y rechazar la hipótesis de ruido i. i. d. si mas de dos o tres de cada 40 quedan fuera de los limites $\pm 1.96/\sqrt{n}$ o si uno está muy por fuera de los límites.

2.8. Modelo ARIMA para series de tiempo no estacionarias

Definición 2.8.1. Si d es un entero no negativo, entonces $\{X_t : t \in T\}$ se dice que es un ***proceso*** ARIMA(p,d,q) si

$$Y_t := (1-B)^d X_t$$

es un proceso ARMA(p,q) causal.

De la definición anterior, se afirma que $\{X_t : t \in T\}$ satisface la siguiente ecuación de diferencia

$$\phi^*(B)X_t \equiv \phi(B)(1-B)^d X_t = \theta(B)Z_t, \text{ donde } \{Z_t\} \sim \text{WN}(0,\sigma^2), \tag{2.50}$$

y $\phi(z)$, $\theta(z)$ son polinomios de grado p y q respectivamente y además $\phi(z) \neq 0$ para $|z| \leq 1$. Por otra parte, el polinomio $\phi^*(z)$ tiene un cero de orden d en $z = 1$ además se observa de que el proceso $\{X_t : t \in T\}$ es estacionario si y sólo si $d = 0$, el cual se reduce a un proceso ARMA(p,q).

Notese que si $d \geq 1$, se puede incluir un polinomio de tendencia arbitrario de grado $(d-1)$ a $\{X_t : t \in T\}$, el cual no sufre cambio alguno a la ecuación de diferencia (2.50). De

esto último se tiene que los procesos ARIMA son usados para poder representar datos que tengan algún tipo de tendencia, sin embargo, cabe destacar también que estos procesos son apropiados para el modelado de series los cuales no tengan tendencia alguna, exceptuando el caso cuando se tiene $d = 0$, en el cual la media de $\{X_t : t \in T\}$ no es posible determinarla por (2.50). Dado que $d \geq 1$, (2.50) determina propiedades de segundo orden para $\{(1-B)^d X_t\}$ pero no para X_t luego los estimadores ϕ, θ y σ^2 estarán basados en las diferencias $(1-B)^d X_t$.

2.8.1. Identificación de técnicas

1. **Transformaciones preliminares**. Al estimar los valores p, q de un modelo ARMA(p, q) para una serie de datos, es poco aceptable pensar que los datos sean una realización de un proceso ARMA, en particular una realización de un proceso estacionario. Si los datos muestran características de las cuales da a sugerir la no estacionalidad de la serie (como puede ser, tendencia y estacionalidad) entonces es necesario el realizar alguna transformación para poder obtener una nueva serie la cual cumpla con la condición de que sea estacionaria.

   La gráfica de la serie o la función de autocorrelación muestral o ambos pueden ser ocupados para poder determinar la estacionaridad de la serie. Al inspeccionar la gráfica de la serie ocasionalmente revela una gran dependencia de variabilidad sobre el nivel de la serie, en tal caso los datos deberán primero ser transformados para reducir o eliminar dicha dependencia. Una de estas transformaciones es la logarítmica $V_t = \ln U_t$ la cual es apropiada cuando $\{U_t\}$ es una serie cuya desviación estándar incrementa linealmente junto con la media. Para una clase general en donde se trata de estabilizar la varianza una transformación importante es la llamada trasformación de *Box-Cox* que se define mediante:

$$f_\lambda(U_t) = \begin{cases} \lambda^{-1}\left(U_t^\lambda - 1\right), & U_t \geq 0,\ \lambda > 0; \\ \ln U_t, & U_t > 0,\ \lambda = 0. \end{cases}$$

   En la practica, el valor que toma λ cuando se aplica esta transformación es $\lambda = 0$ o $\lambda = 0.5$.

   Para poder eliminar la tendencia o la estacionalidad puede ser usado uno de los tres métodos que se describen a continuación:

   a) *Descomposición clásica*

   Descomposición de la serie en componente de tendencia, componente de estacionalidad y componente aleatoria.

   b) *Diferenciación*

   Después de haber eliminado las componentes de estacionaridad y de tendencia, es posible que la función de autocorrelación sea la de una proceso que no sea

estacionario (o cercano a la de uno no estacionario), en tal caso la diferenciación puede ser aplicable.

c) Armónicos

Ajustar una suma de armónicos o un polinomio de tendencia que genere una secuencia de ruido el cual consista de los residuos de la regresión.

2. **Identificación y estimación**. El problema ahora es encontrar el modelo ARMA(p, q) el cual represente a la serie $\{X_t : t \in T\}$. Si p, q son conocidos, simplemente se aplicarían las tecnicas ya planteadas anteriormente sin embargo, esto no es el caso y se hace necesario el identificar los valores apropiados para p y q.

   Para poder elegir p y q se basará principalmente en la minimización del estadístico AICC, el cual esta definido como

$$\text{AICC}(\phi, \theta) = -2 \ln L \left(\phi, \theta, \frac{S(\phi, \theta)}{n} \right) + \frac{2n(p+q+1)}{n-p-q-2}, \tag{2.51}$$

   donde $L(\phi, \theta, \sigma^2)$ es la verosimilitud de los datos encontrados en el ARMA gaussiano con parámetros (ϕ, θ, σ^2) y $S(\phi, \theta)$ es la suma residual cuadrática el cual se definió en (2.45). Una vez que se halla minimizado el valor de AICC entonces será necesario verificar el modelo por medio de la bondad de ajuste, es decir, se verifica que los residuos se comporten como ruido blanco.

   Para cualesquiera valores fijos p, q, los estimadores de máxima verosimilitud de ϕ y θ son los valores que minimizan el estadístico AICC y por tanto, el modelo mínimo AICC puede ser encontrado (sobre cualquier valor de p y q) calculando los estimadores de máxima verosimilitud para cada valor fijo p, q y eligiendo de estos el modelo de máxima verosimilitud con el valor más pequeño de AICC.

2.9. Raíces unitarias en modelos de series de tiempo

El problema de raíces unitarias en series de tiempo surgen cuando cualquier polinomio autorregresivo o de promedios móviles de un modelo ARMA tiene una raíz en o cerca del circulo unitario, puesto que una raíz unitaria en cualquiera de estos polinomios tiene una importante implicación en el modelado. Por ejemplo, una raíz del polinomio autorregresivo cercano al 1 sugiere que los datos deberán diferenciarse antes de poder ajustar un modelo ARMA. Por otra parte, al considerar raíces cercanas a la unidad en un polinomio de promedios móviles da un indicativo de que los datos han sido sobrediferenciado (Véase [3]).

2.9.1. Raíces unitarias en polinomios autorregresivos

En esta parte se discutirá el uso del operador diferencial en una serie no estacionaria cuya función de autocorrelación muestral decae lentamente. El grado de diferenciación de una

serie de tiempo $\{X_t : t \in T\}$ está determinado en gran parte por la aplicación sucesiva de dicho operador hasta que la función de autocorrelación muestral de $\{\nabla^d X_t\}$ decrezca rápidamente. La serie de tiempo diferenciada podría entonces modelar un proceso ARMA(p, q) de orden pequeño y entonces resultar ser un modelo ARIMA(p, d, q) para los datos originales cuyo polinomio autorregresivo esta dado por $(1 - \phi_1 z - \cdots - \phi_p z^p)(1 - z)^d$ con d raíces sobre el circulo unitario.

Se estudia primero el caso cuando se tiene un modelo AR(1), posteriormente el caso general, AR(p). Sea $X_1, X_2, \ldots, X_n$ observaciones de un modelo AR(1)

$$X_t - \mu = \phi_1(X_{t-1} - \mu) + Z_t, \qquad \{Z_t\} \sim \mathrm{WN}(0, \sigma^2), \tag{2.52}$$

donde $|\phi_1| < 1$ y $\mu = \mathrm{E}[X_t]$. Para n lo suficientemente grande el estimador de máxima verosimilitud $\hat{\phi}$ de ϕ es aproximadamente $\mathrm{N}(\phi_1, (1-\phi_1^2)/n)$. Para el caso de raíces unitarias, esta aproximación normal no es convincente y además, no es aplicable el uso de prueba de hipótesis para raíces unitarias $H_0 : \phi_1 = 1$ contra $H_1 : \phi < 1$. Luego para poder construir una prueba de hipótesis para H_0, el modelo (2.52) se puede reescribir como

$$\nabla X_t = X_t - X_{t-1} = \phi^* + \phi_1 X_{t-1} + Z_t, \ \text{ com } \ \{Z_t\} \sim \mathrm{WN}(0, \sigma^2), \tag{2.53}$$

donde

$$\phi_0^* = \mu(1 - \phi_1); \qquad \phi_1^* = \phi_1 - 1.$$

Sea $\hat{\phi}_1^*$ el estimador de mínimos cuadrados de ϕ_1^* encontrado por regresión de ∇X_t sobre 1 y X_{t-1}.

El error estimado estándar de $\hat{\phi}_1^*$ está definido mediante:

$$\widehat{\mathrm{SE}}(\hat{\phi}_1^*) = S \left(\sum_{t=2}^{n} \left(X_{t-1} - \bar{X} \right)^2 \right)^{-1/2},$$

donde

$$S^2 = \sum_{t=2}^{n} \left(\frac{\left(\nabla X_t - \hat{\phi}_0^* - \hat{\phi}_1^* X_{t-1} \right)^2}{n-3} \right),$$

y $\bar{X}$ es la media muestral de $X_1, \ldots, X_{n-1}$. Dickey y Fuller (Véase [3]) mostraron la distribución límite cuando $n \to \infty$ de la relación t

$$\hat{\tau}_\mu := \frac{\hat{\phi}_1^*}{\widehat{\mathrm{SE}}\left(\hat{\phi}_1^* \right)}, \tag{2.54}$$

bajo el supuesto de raíces unitarias $\phi_1^* = 0$, por la que una prueba de la hipótesis nula $H_0 : \phi_1 = 1$ puede ser construida.

El procedimiento anterior puede ser extendida para el caso en donde $\{X_t : t \in T\}$ sigue un modelo AR(p) con media μ dado por

$$X_t - \mu = \phi_1(X_{t-1} - \mu) + \cdots + \phi_p(X_{t-p} - \mu) + Z_t, \qquad \{Z_t\} \sim \text{WN}(0, \sigma^2).$$

El modelo anterior, puede ser reescrito como

$$\nabla X_t = \phi_0^* + \phi_1^* X_{t-1} + \phi_2^* \nabla X_{t-1} + \cdots + \phi_p^* \nabla X_{t-p+1} + Z_t, \tag{2.55}$$

$$\phi_0 = \mu(1 - \phi_1 - \cdots - \phi_p), \quad \phi_1^* = \sum_{i=1}^{p} \phi_i - 1, \quad \phi_j^* = -\sum_{i=j}^{p} \phi_i,\ j = 2, \ldots, p\ .$$

Si el polinomio autorregresivo tiene una raíz en 1, entonces $0 = \phi(1) = -\phi_1^*$ y la serie diferenciada $\{\nabla X_t\}$ es un proceso ARMA($p-1$). Por consiguiente, la prueba de hipótesis de raíz unitaria en 1 para el polinomio autorregresivo es equivalente a la prueba de $\phi_1^* = 0$. Como ocurrió en el caso del proceso AR(1), ϕ_1^* puede ser estimado como el coeficiente de X_{t-1} por la regresión de mínimos cuadrados de ∇X_t sobre $1, X_{t-1}, \nabla X_{t-1}, \ldots, \nabla X_{t-p+1}$. Para n suficientemente grande, la proporción t

$$\hat{\tau}_\mu := \frac{\hat{\phi}_1^*}{\widehat{\text{SE}}\left(\hat{\phi}_1^*\right)}, \tag{2.56}$$

donde $\widehat{\text{SE}}\left(\hat{\phi}_1^*\right)$ es el error estándar estimado de $\hat{\phi}_1^*$ que tiene la misma distribución límite como el estadístico de prueba en (2.54).

2.9.2. Raíces unitarias en polinomios de promedios móviles

Una raíz unitaria en polinomios de promedios móviles tiene cabida un gran número de interpretaciones el cual depende del modelo que sea utilizado. Supongase el caso de que $\{X_t : t \in T\}$ es un proceso ARMA causal e invertible el cual satisface las ecuaciones,

$$\phi(B)X_t = \theta(B)Z_t, \qquad \{Z_t\} \sim \text{WN}(0, \sigma^2).$$

Entonces, la serie diferenciada $Y_t := \nabla X_t$ es un proceso ARMA($p, q+1$) no invertible cuyo polinomio de promedio móvil es $\theta(z)(1-z)$. Por consiguiente, al realizar una prueba de hipótesis sobre raíces unitarias en los polinomios de promedios móviles es equivalente a realizar una prueba de sobrediferenciación en la serie de tiempo.

Como segunda aplicación es posible el distinguir entre los modelos

$$\nabla^k X_t = a + V_t \qquad \text{y} \qquad X_t = c_0 + c_1 t + \cdots + c_k t^k + W_t$$

donde $\{V_t\}$ y $\{W_t\}$ son procesos ARMA invertibles. Para el modelo de la serie diferenciada $\{\nabla^k X_t\}$ no tiene raíces unitarias de promedios móviles, mientras que el segundo modelos

$\{\nabla^k X_t\}$, tiene múltiples raíces de promedios móviles de orden k. Así, se puede distinguir entre dos modelos observando los valores de $\{\nabla^k X_t\}$ para detectar la presencia de raíces unitarias de promedios móviles.

Como primer caso, se estudiará el modelo MA(1), puesto que el caso general es mucho más complicado y no es fácil de resolver. Sea $X_1, \ldots, X_n$ observaciones del modelo MA(1)

$$X_t = Z_t + \theta Z_{t-1}, \qquad \{Z_t\} \sim \text{IID}(0, \sigma^2).$$

Davis y Dunsmuir (Véase [3]) mostraron bajo el supuesto de $\theta = 1$, $n(\hat{\theta} + 1)$ converge en distribución donde $\hat{\theta}$ es el estimador de máxima verosimilitud. Una prueba de hipótesis con $H_0 : \theta = -1$ vs. $H_1 : \theta > -1$ puede ser mostrado limitándose al rechazar H_0 cuando

$$\hat{\theta} > -1 + \frac{c_\alpha}{n},$$

donde c_α es el cuantil $(1 - \alpha)$ de la distribución límite de $n(\hat{\theta} + 1)$.

2.9.3. Predicción para modelos ARIMA

Suponga que la serie $\{Y_t : t \in T\}$ es un proceso ARMA(p, q) causal y que X_0 es una variable aleatoria arbitraria. Defina

$$X_t = X_0 + \sum_{j=1}^{t} Y_j.$$

Se tiene que $\{X_t : t \in T\}$ es una proceso ARIMA$(p, 1, q)$ con media $\text{E}(X_t) = \text{E}(X_0)$ y función de autocovarianza $\text{E}(X_{t+h}X_t) - \text{E}(X_0)^2$ el cual sólo depende de $\text{Var}(X_0)$ y de $\text{Cov}(X_0, Y_j)$ para $j = 1, 2, \ldots$.

En efecto, se tiene que para cada $t = 1, 2, \ldots$ se observa que,

$$Y_t = (1 - B)X_t.$$

Puesto que $\{Y_t : t \in T\}$ es un proceso ARMA(p, q), se tiene que

$$\phi(B)Y_t = \theta(B)Z_t,$$

así, al sustituir el valor de Y_t en términos de X_t, se obtiene que

$$\phi(B)(1 - B)X_t = \theta(B)Z_t,$$

por definición, se tiene que $\{X_t, t \geq 0\}$ es un proceso ARIMA$(p, 1, q)$. Además,

$$\begin{aligned}\mathrm{E}[X_t] &= \mathrm{E}\left[X_0 + \sum_{j=1}^{t} Y_t\right] \\ &= \mathrm{E}[X_0] + \sum_{j=1}^{t} \mathrm{E}[Y_t] \\ &= \mathrm{E}[X_0],\end{aligned}$$

la última igualdad se da, puesto que se tiene que $\{Y_t : t \in T\}$ es un proceso causal. Por otra parte, para el cálculo de la función de autocovarianza se tiene:

$$\begin{aligned}\mathrm{Cov}(X_{t+h}, X_t) &= \mathrm{E}\left[\left(X_{t+h} - \mathrm{E}[X_{t+h}]\right)\left(X_t - \mathrm{E}[X_t]\right)\right] \\ &= \mathrm{E}\left[\left(X_{t+h} - \mathrm{E}[X_0]\right)\left(X_t - \mathrm{E}[X_0]\right)\right] \\ &= \mathrm{E}[X_{t+h}X_t] - (\mathrm{E}[X_0])^2 \\ &= \gamma_X(h).\end{aligned}$$

El mejor predictor lineal de X_{n+1} basado en $\{1, X_0, X_1, \ldots, X_n\}$ es el mismo que el mejor predictor lineal en términos de $\{1, X_0, Y_1, \ldots, Y_n\}$ puesto que cada combinación lineal de este último es combinación lineal de la primera y viceversa. Así, denotando como P_n al mejor polinomio predictor en términos de su respectivo conjunto y usando el hecho de la linealidad de P_n, entonces se obtiene

$$\begin{aligned}P_n X_{n+1} &= P_n(X_0 + Y_1 + \cdots + Y_{n+1}) \\ &= P_n(X_n + Y_{n+1}) \\ &= X_n + P_n Y_{n+1}.\end{aligned}$$

Para el caso general, se asumirá que el proceso observado $\{X_t : t \in T\}$ satisface la ecuación de diferencia

$$(1 - B)^d X_t = Y_t, \quad t = 1, 2, \ldots ,$$

donde $\{Y_t : t \in T\}$ es un proceso ARMA(p, q) causal y además, el vector aleatorio $(X_{1-d}, \ldots, X_0)$ es no correlacionado respecto a Y_t para $t > 0$. La ecuación de diferencia anterior, puede ser reescrita de la siguiente forma

$$X_t = Y_t - \sum_{j=1}^{d} \binom{d}{j} (-1)^j X_{t-j}, \quad t = 1, 2 \ldots . \tag{2.57}$$

Para poder realizar el cálculo de P_nX_{n+h}, se aplica a ambos miembros de (2.57) el operador P_n para obtener

$$P_nX_{n+h} = P_nY_{n+h} - \sum_{j=1}^{d} \binom{d}{j} (-1)^j P_nX_{n+h-j}. \tag{2.58}$$

Teniendo el supuesto de que $(X_{1-d}, \ldots, X_0)$ es no correlacionado respecto a Y_t con $t > 0$, esto permite identificar P_nY_{n+h} como el mejor predictor lineal de Y_{n+h} en terminos de $\{1, Y_1, \ldots, Y_n\}$. Por otra parte, P_nX_{n+1} se obtiene directamente de (2.58) haciendo notar que $P_nX_{n+1-j} = X_{n+1-j}$ para $j \geq 1$. El predictor P_nX_{n+2} puede ser encontrado de (2.58) una vez calculado el valor de P_nX_{n+1}; los predictores P_nX_{n+3}, $P_nX_{n+4}, \ldots$ son calculados recursivamente de la misma manera.

Para hallar el error medio cuadrático del predictor es conveniente el expresar P_nY_{n+h} en terminos de $\{X_j\}$. Para $n \geq 0$, se denota el predictor de un paso mediante $\hat{Y}_{n+1} = P_nY_{n+1}$ y $\hat{X}_{n+1} = P_nX_{n+1}$. Así de (2.57) y (2.58) se obtiene,

$$X_{n+1} - \hat{X}_{n+1} = Y_{n+1} - \hat{Y}_{n+1}, \quad n \geq 1,$$

así de (2.37) si $n > m = \max(p, q)$ y $h \geq 1$, se obtiene que,

$$P_nY_{n+h} = \sum_{i=1}^{p} \phi_i P_nY_{n+h-i} + \sum_{j=h}^{q} \theta_{n+h-1,j} \left(X_{n+h-j} - \hat{X}_{n+h-j} \right). \tag{2.59}$$

Ajustando $\phi^*(z) = (1-z)^d\phi(z) = 1 - \phi_1^* z - \cdots - \phi_{p+d}^* z^{p+d}$, se encuentra de (2.58) y (2.59) que

$$P_nX_{n+h} = \sum_{j=1}^{p+d} \phi_j^* P_nX_{n+h-j} + \sum_{j=h}^{q} \theta_{n+h-1,j} \left(X_{n+h-j} - \hat{X}_{n+h-j} \right), \tag{2.60}$$

la cual es análoga a la fórmula de predicción en h pasos (2.37) para un proceso ARMA.

El error medio cuadrático para el predictor en h-pasos esta dado por:

$$\sigma^2(h) = \mathrm{E}(X_{n+h} - P_nX_{n+h})^2 = \sum_{j=0}^{h-1} \left(\sum_{r=0}^{j} \chi_r \theta_{n+h-r-1,j-r} \right)^2 v_{n+h-j-1}, \tag{2.61}$$

done $\theta_{n0} = 1$, además

$$\chi(z) = \sum_{r=0}^{\infty} \chi_r z^r = \left(1 - \phi_1^* z - \cdots - \phi_{p+d}^* z^{p+d} \right)^{-1},$$

y

$$v_{n+h-j-1} = \mathrm{E}\left(X_{n+h-j} - \hat{X}_{n+h-j}\right)^2 = \mathrm{E}\left(Y_{n+h-j} - \hat{Y}_{n+h-j}\right)^2.$$

Los coeficientes χ_j pueden ser hallados de manera recursiva mediante (2.38)con ϕ_j^* reemplazado por ϕ_j. Para n suficientemente grande, se puede aproximar (2.61), siempre y cuando $\theta(\cdot)$ sea invertible, mediante:

$$\sigma_n^2(h) = \sum_{j=0}^{h-1} \psi_j^2 \sigma^2, \tag{2.62}$$

donde

$$\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = (\phi^*(z))^{-1}\theta(z).$$

2.9.4. Función predictora

Analizando (2.60) se muestra que para $n > m = \text{máx}(p, q)$, el predictor de $h-$ pasos

$$g(h) := P_n X_{n+h},$$

satisface las ecuaciones diferenciales lineales homogéneas

$$g(h) - \phi_1^* g(h-1) - \cdots - \phi_{p+d}^* g(h-p-d) = 0, \quad h > q, \tag{2.63}$$

donde $\phi_1^*, \ldots, \phi_{p+d}^*$ son los coeficientes de $z, \ldots, z^{p+d}$ en

$$\phi^*(z) = (1-z)^d \phi(z).$$

Si se asume que los ceros de $\phi(z)$ (denotado por $\xi_1, \xi_2, \ldots, \xi_p$) son todos distintos entonces la solución de (2.63) estará dada por

$$g(h) = a_0 + a_1 h + \cdots + a_d h^{d-1} + b_1 \xi_i^{-h} + \cdots + b_p \xi_p^{-h}, \quad h > q - p - d, \tag{2.64}$$

donde los coeficientes $a_1, \ldots, a_d$ y $b_1, \ldots, b_p$ son determinados por las $p + d$ ecuaciones obtenidas al igualar el lado derecho de (2.64) para $q-p-d < h \leq q$ con los correspondientes valores de $g(h)$ calculados numéricamente. Una vez que se hallan evaluado las constantes a_i y b_i la expresión algebraica (2.64) dará los predictores para todo $h > q - p - d$. En el caso de que $q = 0$, los valores de $g(h)$ para $a_0, \ldots, a_d, b_1, \ldots, b_p$ son los valores observados $g(h) = X_{n+h}$ con $-p-d \leq h \leq 0$ y la expresión (2.62) corresponde el error medio cuadrático el cual será exacto.

2.10. Modelos estacional ARIMA

Definición 2.10.1. Si d y D son enteros no negativos, entonces $\{X_t : t \in T\}$ es un ***proceso*** ARIMA **estacional** (SARIMA) ***de orden*** $(p, q, d) \times (P, D, Q)$ ***con periodo*** s, si la serie diferenciada $Y_t = (1 - B)^d(1 - B^s)^D X_t$ es un proceso ARMA causal definido por:

$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)Z_t, \qquad \{Z_t\} \sim \mathrm{WN}(0, \sigma^2), \tag{2.65}$$

donde $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$, $\Phi(z) = 1 - \Phi_1 z - \cdots - \Phi_p z^q$, $\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$ y $\Theta(z) = 1 + \Theta_1 z + \cdots + \Theta_Q z^Q$.

Nota 2.10.1. En aplicaciones D es a lo más uno y P, Q comúnmente son menores que tres.

Nota 2.10.2. (2.65) puede ser reescrito en forma equivalente a

$$\phi^*(B)Y_t = \theta^*(B)Z_t, \tag{2.66}$$

donde $\phi^*(\cdot), \theta^*(\cdot)$ son polinomios de grado $p+sP$ y $q+sQ$ respectivamente, cuyos coeficientes pueden ser expresados en términos de $\phi_1, \ldots, \phi_p$, $\Phi_1, \ldots, \Phi_P$, $\theta_1, \ldots, \theta_q$ y $\Theta_1, \ldots, \Theta_Q$. Bajo la condición de que $p < s$ y $q < s$ las restricciones sobre los coeficientes de $\phi^*(\cdot)$ y $\theta^*(\cdot)$ pueden ser expresados como una relación multiplicativa

$$\phi^*_{is+j} = \phi^*_{is}\phi^*_j, \quad i = 1, 2, \ldots;\ j = 1, \ldots, s-1$$

y

$$\theta^*_{is+j} = \theta^*_{is}\theta^*_j, \quad i = 1, 2, \ldots;\ j = 1, \ldots, s-1\ .$$

El primer paso para la identificación de modelos SARIMA para un conjunto de datos es hallar d y D el cual haga que las observaciones diferenciadas

$$Y_t = (1 - B)^d(1 - B^s)^D X_t,$$

sean aparentemente estacionaria. Después se examina la función de autocorrelación muestral y la función de autocorrelación parcial de $\{Y_t : t \in T\}$ en retrasos que son múltiplos de s para poder identificar los ordenes de P y Q en el modelo (2.66). Si $\hat{\rho}(\cdot)$ es la función de autocorrelación muestral de $\{Y_t : t \in T\}$, entonces P y Q deberán elegirse de tal forma que $\hat{\rho}(ks)$ para $k = 1, 2, \ldots$ sea similar con la función de autocorrelación de un proceso ARMA(P, Q). El orden de p y q son seleccionados de forma que $\hat{\rho}(1), \ldots, \hat{\rho}(s-1)$ coincida con la función de autocorrelación de un proceso ARMA(p, q). Por último, el criterio AICC y la prueba de bondad de ajuste son usados para seleccionar el mejor modelo SARIMA.

Para valores dados de p, d, q, P, D y Q, los parámetros $\phi, \theta, \mathbf{\Phi}, \mathbf{\Theta}$ y σ^2 pueden ser encontrados usando el procedimiento de máxima verosimilitud. Las diferencias $Y_t = (1 - B)^d(1 - B^s)^D X_t$ constituye un proceso ARMA$(p + sP, q + sQ)$ el cual algunos de los coeficiente son cero y el resto son funciones del vector $(p + q + Q)$ dimensional

$\beta' = (\phi', \Phi', \theta', \Theta')$. La máxima verosimilitud de β es el valor que minimiza $\ell(\beta)$, y la máxima verosimilitud estimado de σ^2 esta dado por (2.44).

Un enfoque más directo para el modelado de series diferenciadas $\{Y_t : t \in T\}$ es simplemente ajustar a un subconjunto el modelo ARMA de la forma (2.66) sin hacer uso de los multiplicadores que son de la forma $\phi^*(\cdot)$ y $\theta^*(\cdot)$ en (2.65).

2.10.1. Proceso de predicción en el modelo SARIMA

El proceso de predicción en modelos SARIMA es análoga para la predicción de modelos ARIMA (Véase sección 2.9.3, página 43). Expandiendo el operador $(1-B)^d(1-B^s)^D$ en potencias de B y reordenando la ecuación

$$(1-B)^d(1-B^s)^D X_t = Y_t,$$

ajustado $t = n+h$ se tendrá

$$X_{n+h} = Y_{n+h} + \sum_{j=1}^{d+Ds} a_j X_{n+h-j} \tag{2.67}$$

el cual es análogo a (2.58). Haciendo el supuesto de que las primeras $d+Ds$ observaciones $X_{-d-Ds+1}, \ldots, X_0$ son no correlacionados con $\{Y_t,\ t \geq 1\}$, se puede determinar el mejor predictor lineal $P_n X_{n+h}$ de X_{n+h} basado en $\{1, X_{-d-Ds+1}, \ldots, X_n\}$ aplicando P_n a cada lado de (2.67) para obtener

$$P_n X_{n+h} = P_n Y_{n+h} + \sum_{j=1}^{d+Ds} a_j P_n X_{n+h-j}, \tag{2.68}$$

El primer término del lado derecho coincide con el mejor predictor lineal del proceso ARMA $\{Y_t : t \in T\}$ en términos de $\{1, Y_1, \ldots, Y_n\}$. El predictor $P_n X_{n+h}$ puede ser calculado de forma recursiva para $h = 1, 2, \ldots$ de (2.68), si es que se cumple que $P_n X_{n+h} = X_{n+1-j}$ para cada $j \geq 1$.

Un argumento similar dado en (2.61) nos proporciona el error cuadrático medio

$$\sigma_n^2(h) = \mathrm{E}[X_{n+h} - P_n X_{n+h}]^2 = \sum_{j=0}^{h-1} \left(\sum_{r=0}^{j} \chi_r \theta_{n+h-r-1,j-r} \right)^2 v_{n+h-j-1}, \tag{2.69}$$

donde θ_{nj} y v_n son obtenidos al aplicar el algoritmo innovations a la serie diferenciada $\{Y_t : t \in T\}$ y

$$\chi(z) = \sum_{r=0}^{\infty} \chi_r z^r = \left[\phi(z)\Phi(z^s)(1-z)^d(1-z^s)^D \right]^{-1}, \quad |z| < 1. \tag{2.70}$$

Para n lo suficientemente grande se puede aproximar (2.69) si $\theta(z)\Theta(z^s)$ es no cero para todo $|z| \leq 1$ por

$$\sigma_n^2(h) = \sum_{j=0}^{h-1} \psi_j^2 \sigma^2, \tag{2.71}$$

donde

$$\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{\theta(z)\Theta(z^s)}{\phi(z)\Phi(z^s)(1-z)^d(1-z^s)^D}, \quad |z| < 1.$$

En este Capítulo se desarrolló la teoría estadística necesaria para realizar pronóstico a un conjunto de datos, el cual conlleva (de manera general), a una serie de pasos tales como: preparación de los datos, propuestas de modelos, elección de modelos y como último punto la realización de la predicción (Véase [3]).

Se han desarrollado nuevos enfoques para la realización de pronósticos. Uno de estos son las redes neuronales artificiales, cuyo algoritmo de aprendizaje *backpropagation* ha sido empleado en muchas áreas, incluyendo en medicina en el diagnostico de enfermedades (Véase [12]) además de realizar modelo para la predicción de los indices de precios de las bolsas bursátiles de un país o estimar el desempeño de una empresa (Véase [9]) y el de la cancelación de ruido a través de cables de transmisión, como por ejemplo, la linea telefónica (Véase [14]).

Capítulo 3

Redes neuronales artificiales y biológicas

En el Capítulo anterior se presentó la teoría necesaria para realizar la predicción usando series de tiempo el cual es necesario que el conjunto de datos cumplan cierta hipótesis para que, en primer lugar, se busque un modelo estadístico que se ajuste a éstos de manera adecuada y posteriormente realizar la predicción.

En este Capítulo se presenta la teoría para el desarrollo de redes neuronales artificiales cuya rama de la inteligencia artificial se destaca de manera impresionante al resolver problemas de cierto índole como es el caso del pronóstico.

3.1. Redes neuronales biológicas

Varios investigadores han desarrollado desde hace más de 30 años la teoría de redes neuronales artificiales quienes se basaron en la eficiencia y características de los procesos llevados a cabo por el cerebro (Véase [5]), tales como:

1. *Tolerante a fallas*. En el cerebro humano mueren diariamente células sin afectar su funcionamiento.

2. *Flexibilidad*. Se ajusta a nuevos ambientes por medio de un proceso de aprendizaje. No hay que programarlo.

3. *Tratamiento de incertidumbre*. Puede manejar información difusa, con ruido o información inconsistente.

4. *Multiproceso*. Es altamente paralelo, esto en el sentido de que el cerebro procesa cierta información en forma paralela. Por ejemplo, se puede caminar y al mismo tiempo observar el paisaje además de escuchar el canto de las aves todo al mismo tiempo.

De forma muy general, una ***red neuronal*** es una máquina que modela la forma en que el cerebro realiza una tarea en concreta. Para lograr esto, y tener un buen rendimiento, la

red neuronal emplea interconecciones de manera masiva entre elementos simples de cómputo llamados ***neuronas***.

Definición 3.1.1. Una red neuronal es un procesador masivamente distribuido paralelo compuesto de unidades simples de procesamiento los cuales tienen una propensión natural para el almacenamiento de conocimiento y hacer disponible su uso.

La red neuronal se asemeja al cerebro en dos aspectos principales:

1. El conocimiento es adquirido por la red a través de un proceso de aprendizaje.
2. Los fuerza de conexión entre las neuronas, llamado ***pesos sinapticos***, son usados para el almacenamiento del conocimiento adquirido.

Las neuronas están constituidas principalmente de tres partes, (ver Figura 3.1), ***dendritas***, el ***cuerpo de la neurona*** y el ***axón***. Las dendritas son el receptor de la neurona, son ramificaciones de fibras nerviosas las cuales recibe las señales eléctricas de otras neuronas y ésta pasa a través del cuerpo de la neurona que se encarga de la suma de estas señales de entradas. El axón es una fibra larga cuyo objetivo es llevar la señal desde el cuerpo de la neurona hacia otras neuronas mediante el contacto de las dendrita de éstas. El punto de contacto recibe el nombre de ***sinapsis***.

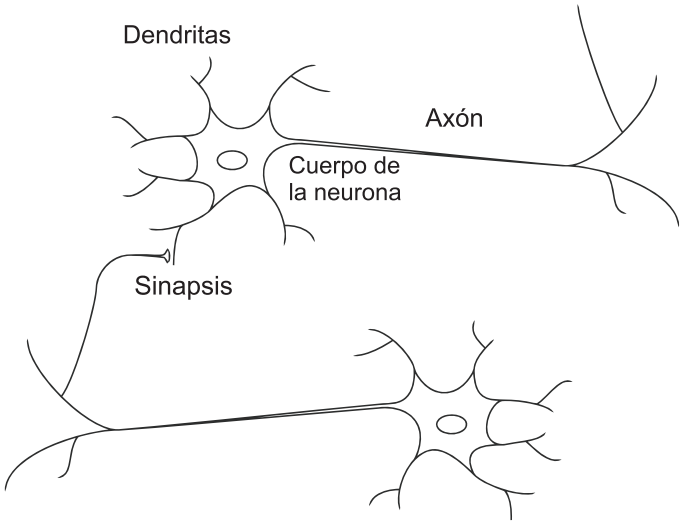


Figura 3.1: Partes principales de la neurona biológica.

3.2. Modelo neuronal artificial

Se identifica tres elementos básicos de un modelo neuronal artificial (Véase [10]) las cuales son:

1. Un conjunto de *sinapsis* o *enlaces de conexión*. Cada una está caracterizada por un *peso* o *fuerza*. Una señal de entrada p_j de la sinapsis j está conectado con la neurona k la cual es multiplicado por el peso sinaptico w_{kj}.

2. Un *sumador* o *combinación lineal*. Combina las señales de entradas ponderandolas por las respectivas sinapsis de la neurona.

3. Una *función de activación* f, el cual limita o normaliza la amplitud de la salida de la neurona.

En la Figura 3.2, se muestra un ejemplo de un modelo neuronal. Observe que en este modelo incluye un ***sesgo*** o ***ganancia*** b_k que es aplicado desde el exterior de la neurona. Esta ganancia tiene el efecto de incrementar o disminuir la entrada neta de la función de activación.

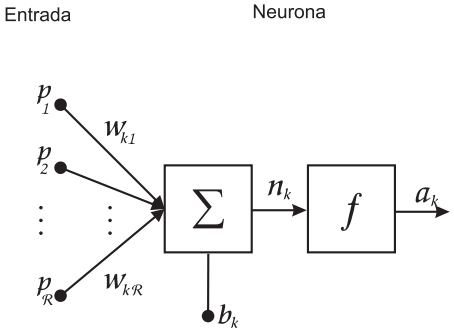


Figura 3.2: Modelo Neuronal

En términos matemáticos, la descripción del modelo de una neurona k puede expresarme mediante las siguientes ecuaciones:

$$u_k = \sum_{j=1}^{R} w_{kj} p_j \tag{3.1}$$

y

$$a_k = f(u_k + b_k) \tag{3.2}$$

donde $p_1, p_2, \cdots, p_R$ son las señales de entradas; $w_{k1}, w_{k2}, \cdots, w_{kR}$ son los pesos sinapticos de la neurona k; u_k es la *combinación lineal de salida* entre las señales de entrada; b_k es el sesgo o ganancia; f es la *función de activación* y por último a_k es la señal de salida de la neurona. El uso de la ganancia b_k tiene el efecto de aplicar una ***transformación afín*** a la salida u_k de la combinación lineal del modelo mostrado en la Figura 3.2, el cual es llamado *entrada neta* de la red y está dada por:

$$n_k = u_k + b_k \tag{3.3}$$

En particular, dependiendo si la ganancia b_k es positiva o negativa, la relación entre el *campo inducido local* o el *potencial de activación* a_k de la neurona k y de la combinación lineal de la salida u_k es modificado en la forma que se ilustra en la Figura 3.3.

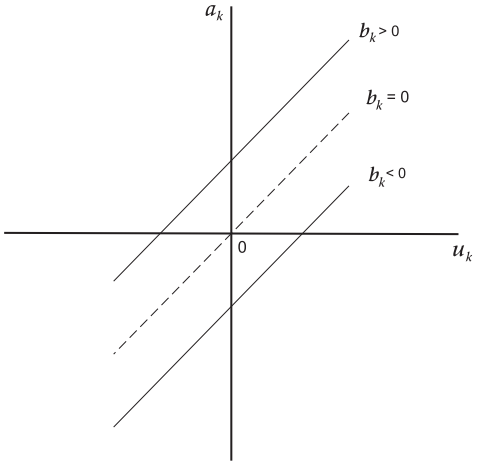


Figura 3.3: Transformación afín producido por la presencia de la ganancia.

3.2.1. Función de transferencia

La función de activación o función de transferencia f define la salida de la neurona en términos de la entrada neta n_k. La elección de esta función dependerá si satisface alguna especificación del problema que se desea resolver.

Existe una gran variedad de funciones de transferencia sin embargo, las siguientes funciones que se describen a continuación son las más usadas.

1. Función de transferencia ***límite estricto*** (*hard limit*) o ***función umbral***. Establece la salida de la neurona como 0 si el argumento es menor que 0 o 1 si su argumento es mayor o igual a 0. Este tipo de función es utilizada para clasificar a la entrada de la neurona en dos categorías diferentes.

$$f(n_k) = \begin{cases} 0, & \text{si } n_k < 0; \\ 1, & \text{si } n_k \geq 1. \end{cases} \tag{3.4}$$

   En la práctica, es aconsejable tener una función de transferencia el cual tome valor de -1 o 1, en tal caso, la función de transferencia *hardlim* se toma como una función simétrica respecto al origen. De aquí que la función *hardlims* se define como la función *hardlim* simétrica respecto al origen, es decir

$$f(n_k) = \begin{cases} -1, & \text{si } n_k < 0; \\ 1, & \text{si } n_k \geq 1. \end{cases} \tag{3.5}$$

2. Función ***lineal***. La salida para esta función de transferencia será igual a la entrada neta.

$$f(n_k) = n_k. \tag{3.6}$$

3. Función ***sigmoidal***. Este tipo de función es comúnmente usada en la construcción de redes neuronales artificiales. Un ejemplo de la función sigmoidal es la función ***log-sigmoidal*** el cual está definida mediante la expresión

$$f(n_k) = \frac{1}{1 + e^{-n_k}} \ . \tag{3.7}$$

Mientras que la función *hardlim* toma valores de 0 o 1 la función sigmoidal toma un rango continuo de valores entre 0 y 1, además la función *hardlim* es una función no diferenciable mientras que la sigmoidal lo es.

4. Función ***tangente sigmoidal***. Esta función está definida como

$$f(n_k) = \frac{e^{n_k} - e^{-n_k}}{e^{n_k} + e^{-n_k}}. \tag{3.8}$$

3.2.2. Arquitectura de la red

De manera general, se identifica tres clases importantes diferentes de arquitectura de una red.

Redes de capa simple con conexiones hacia adelante

Como se mencionó anteriormente (Véase sección 1.3, página 8), una red neuronal *multicapa* es aquella red cuyas neuronas están organizadas por capas. La red más simple de una red multicapa es aquella de una sóla capa, es decir, sólo se tiene la capa de entrada. Aquí se tomará la arquitectura mencionada en [8] el cual distingue la capa de entrada de la señal de entrada. Por ejemplo, en [10] no hace distinción a la capa de entrada con la señal de entrada además de que la única diferencia que realiza es la capa de salida con la señal de salida. Con esto, se tiene que una neurona simple, o neurona *monocapa*, es aquella red con una única capa la cual es tanto como capa de entrada como capa de salida es decir, no existe capa oculta.

En la Figura 3.4 se observa una red neuronal de una sola capa con S neuronas. Notese que cada elemento de la señal de entrada p_j con $0 \leq j \leq R$ está conectado a cada uno de las neuronas.

Se le denomina ***red de conexión hacia adelante*** o ***red de alimentación hacia adelante*** aquella red cuya información fluye dentro de la misma desde la capa de entrada hacia la capa de salida.

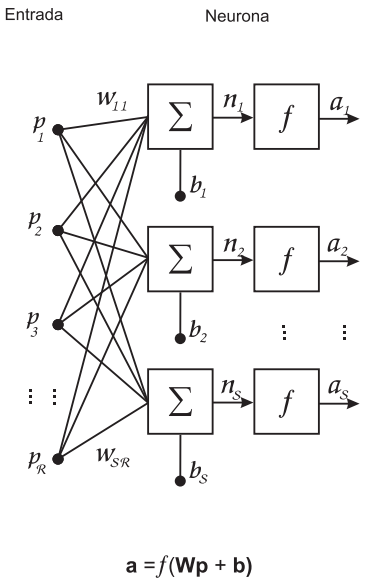


Figura 3.4: Neurona de una capa con S neuronas

Redes multicapa con conexión hacia adelante

Este tipo de redes se distingue de la anterior por la presencia de una o más capas ocultas, las cuales realizan el procesamiento de la señal de entrada de la red.

La señal de entrada en la capa de entrada de la red proporciona respectivamente patrones de activación las cuales constituyen la señal de entrada aplicado a la neurona en la segunda capa (es decir, la primera capa oculta). La señal de salida de la segunda capa es usada como entrada en la tercer capa y así sucesivamente para las demás capas de la red. Las neuronas en cada capa de la red tiene como entrada la señal procedente de la capa inmediata anterior. Luego, el conjunto de señales de salida final en la capa oculta de la red lo constituye de las respuestas totales de los patrones de activación los cuales fueron proporcionados por la señal de entrada en la primera capa.

Para referirse a una red neuronal se auxiliara de los números de neurona que tiene cada capa de la red. Por ejemplo, para poder referirse a la red neuronal de la Figura 3.5 se usará la siguiente notación $R - S^1 - S^2 - S^3$, el cual significa que la red tiene tres capas, la capa de entrada con S^1 neuronas, una capa oculta de S^2 neuronas y la capa oculta de salida de S^3 neuronas. El número de componentes de la señal de entrada corresponde al primer número de la sucesión anterior que para el ejemplo, se tiene una señal de R entradas.

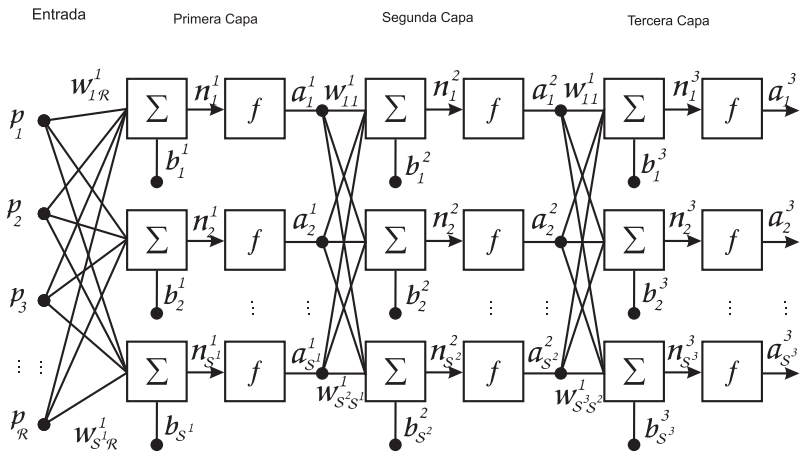


Figura 3.5: Red Multicapa con una sola capa oculta

Redes recurrentes

Este tipo de red se distinguen de las redes con conexión hacia adelante por que existen al menos un bucle o ciclo de retroalimentación. Por ejemplo, una red recurrente puede consistir de una red de una sola capa cuya señal de salida es señal de entrada para las demás neuronas. De manera general este tipo de red es aquella cuya señal de entrada es señal de salida para la red.

3.2.3. Regla de aprendizaje

Por ***regla de aprendizaje*** se entenderá como el procedimiento para la modificación de los pesos y ganancias de una red, además este procedimiento puede ser también referido como algoritmo de entrenamiento. El propósito de la regla de aprendizaje es la de entrenar a la red para que pueda realizar alguna tarea en específica. Los tipos de aprendizaje se dividen en tres categorías: *aprendizaje supervisado*, *aprendizaje no supervisado* y el *aprendizaje semi-supervisado*.

En el ***aprendizaje supervisado***, se es provisto de un conjunto de ejemplos llamados *conjunto de entrenamiento* los cuales es un conjunto de entradas con salidas esperadas de la forma:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\} \tag{3.9}$$

donde $\mathbf{p}_q$ es una entrada o patrón de la red y $\mathbf{t}_q$ es la correspondiente salida esperada llamado *objetivo*. Cada salida que la red proporcione por cada patrón de entrada es comparado con las salidas esperadas, de aquí que la regla de aprendizaje ajusta los pesos y las ganancias de la red de tal forma de que las salidas que proporcione la red sean cercanas a los objetivos esperados.

El ***aprendizaje no supervisado*** no requieren influencia externa para el ajuste de los pesos en las conexiones entre las neuronas, es decir, la neurona no recibe información alguna por parte del entorno que le indique si la salida generada por medio de una entrada es o no la correcta.

Estas redes deben encontrar características, regularidades o categorías que se puedan establecer entre los datos que se presenten en sus entradas. La salida de la red representa en algunos casos, el *grado de familiaridad* o similitud entre la información que se le está presentando a la red y las informaciones que se le han mostrado hasta entonces, estableciendo así categorías.

Por último, el ***aprendizaje semi-supervisado*** realiza la mezcla o la combinación de los aprendizajes antes mencionados.

3.3. Perceptrón simple

La primera red neuronal conocida, fue desarrollada en 1943 por Warren McCulloch y Walter Pitts el cual consistía en la suma de la señal de entrada que eran multiplicadas por ciertos valores de pesos que eran escogidos aleatoriamente. Ésta suma era comparada con un patrón preestablecido el cual determinaba la salida de la red, si la suma era mayor o igual que el patrón preestablecido, la salida de la red es 1, en caso contrario, la salida es cero. Éstos investigadores llegaron a afirmar que cualquier función aritmética o lógica podría ser modelada por esta red. A diferencia de las redes biológicas los parámetros de su red tenía que ser diseñado puesto que aún no existian métodos de aprendizajes en ese tiempo.

En 1950 Frank Rosenblatt y otros investigadores desarrollaron una clase de red neuronal llamado ***perceptrón***. Ésta red era muy similar a la red de McCulloch y Pitts sin embargo, Rosenblatt había introducido la regla de aprendizaje para el entrenamiento de ésta y con ello la red podría resolver problemas de reconocimiento de patrones. Él mostró que la regla de aprendizaje, además de ser simple y automática, siempre convergía a ciertos pesos de los cuales podría dar solución al problema, esto siempre y cuando existieran estos pesos que pudieran resolver el problema.

Desafortunadamente, el modelo del perceptrón tiene limitantes. Estas fueron publicadas en el libro *Perceptrons* por Minsky y Seymour Papert. Ellos demostraron que la red era incapaz de solucionar ciertas problemas llamados *problemas linealmente separables*, los cuales son conjuntos cuyo patrones son linealmente separable. Supongase dos conjuntos de puntos A y B, entonces se dice que son ***linealmente separables*** en un espacio n-dimensional si existen $n+1$ números reales $w_1, \ldots, w_n, \theta$ de manera que para cada punto $(x_1, \ldots, x_n) \in A$ satisface que $\sum_{i=1}^{n} w_i x_i \geq \theta$ y cada punto $(x_1, \ldots, x_n) \in B$ satisface $\sum_{i=1}^{n} w_i x_i < \theta$ (Véase Figura 3.6).

No fue que hasta en 1980 estas limitaciones fueron superadas con la implementación de mejoras en el perceptrón. Algunas de estas mejoras fueron la inclusión de varias capas y de las reglas de aprendizajes asociativos.

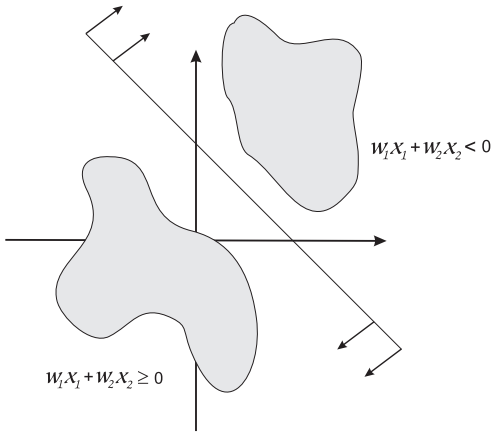


Figura 3.6: Separabilidad lineal en el espacio $\mathbb{R}^2$.

3.3.1. Arquitectura y aprendizaje del perceptrón

El modelo del perceptrón esta *formada de una sola neurona y una o varias señales de entrada*. La función de trasferencia o activación consta de la función *hardlim* o función umbral. El modelo más simple consta de una sola señal de entrada el cual puede ser observado en la Figura 3.7(a)

De acuerdo a (3.2), se tiene que la salida de este modelo esta dado por

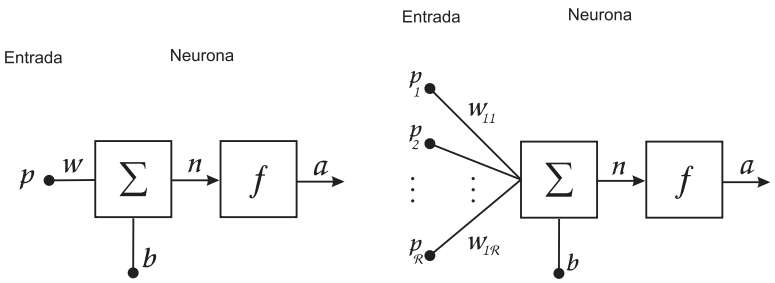
$$a = \begin{cases} 1, & \text{si } n \leq 0; \\ 0, & \text{en otro caso.} \end{cases} \tag{3.10}$$

Observe que los modelos que se ilustran en las Figuras 3.7(a) y 3.7(b), tienen una sola señal de salida. Con esto una red de perceptrón simple se puede clasificar la señal de entrada en dos categorías, ya sea 1 o 0, (esto por la función de transferencia). Por otra parte, en una red de perceptrones de una sola capa con múltiples neuronas, esta puede clasificar la señal de entrada en 2^S posibles categorías; en donde S es el número de neuronas que tiene la red.

Para el modelo mostrado en la Figura 3.7(c) el cual consta de una red neuronal de R entradas y S neuronas (perceptrones) de una sola capa, se define la ***matriz de pesos*** $\mathbf{W}$ asociado a la red neuronal mediante

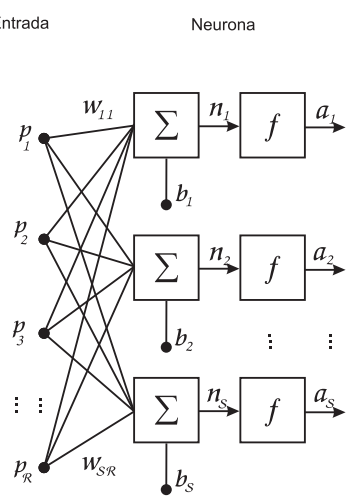
$$\mathbf{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{pmatrix} \tag{3.11}$$

Además se define el vector ${}_i\mathbf{w}$ como el vector columna que esta compuesto de los elementos de la i-ésima fila de la matriz $\mathbf{W}$, es decir:



(a) Una señal de entrada

(b) Varias señales de entrada



a = f(**Wp** + **b**)

(c) Varias neuronas con varias señales de salida

Figura 3.7: Modelo del perceptrón

$$ {}_i\mathbf{w} = \begin{pmatrix} w_{i,1} & w_{i,2} & \cdots & w_{i,R} \end{pmatrix}', \tag{3.12} $$

De esto último, se tiene que ${}_i\mathbf{w}$ consta de los pesos sinápticos de la neurona i con las correspondientes señales de entrada. Luego se tiene que $\mathbf{W}$ puede ser reescrito en términos de ${}_i\mathbf{w}$ de la siguiente manera:

$$ \mathbf{W} = \begin{pmatrix} {}_1\mathbf{w}' & {}_2\mathbf{w}' & \cdots & {}_S\mathbf{w}' \end{pmatrix}'. \tag{3.13} $$

Así, (3.2) queda expresado en forma vectorial de la siguiente manera

$$ \mathbf{a} = f(\mathbf{W}\mathbf{p} + \mathbf{b}) \tag{3.14} $$

donde f es la función umbral, $\mathbf{p}$ es el vector $R \times 1$ definido como

$$\mathbf{p} = \begin{pmatrix} p_1 & p_2 & \cdots & p_R \end{pmatrix}' \tag{3.15}$$

cuya k-ésima entrada de este vector corresponde al k-ésimo peso sináptico de la correspondiente sinapsis k y $\mathbf{b}$ es el vector de ganancia $S \times 1$ el cual cada elemento de este vector es la correspondiente ganancia de la neurona, es decir

$$\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_S \end{pmatrix}' \tag{3.16}$$

Observe que la salida para este tipo de arquitectura es un vector $\mathbf{a}$ de dimensión $S \times 1$ cuya i-ésima salida está dada mediante

$$a_i = {}_i\mathbf{w}'\mathbf{p} + b_i \qquad i = 1, 2, \cdots, S \tag{3.17}$$

Regla de aprendizaje

El perceptrón es una red cuyo tipo de aprendizaje es supervisado. Su comportamiento está definido por los pares de la forma

$$\{\mathbf{p}_1, t_1\}, \{\mathbf{p}_2, t_2\}, \ldots, \{\mathbf{p}_Q, t_Q\}$$

En la regla de aprendizaje se aplica el patrón $\mathbf{p}_q$ a la red el cual proporcionará una salida que es comparada con el valor objetivo o esperado t_q. Los valores de los pesos determinan el funcionamiento de la red y éstos valores se pueden fijar o adoptar con diferentes algoritmos de entrenamiento.

El algoritmo más simple para el entrenamiento de una red tipo perceptrón que consiste de una sola neurona se resume en los siguientes pasos:

1. Se inicializa la matriz de pesos $\mathbf{W}$ y el valor de la ganancia b de forma aleatoria.
2. Se presenta el primer patrón a la red junto con la salida objetivo en forma de pares entrada/salida.
3. Se calcula la salida de la red mediante (3.14)
4. Cuando la red no retorna la salida correcta es necesario alterar el valor de los pesos con el objetivo de que la salida que proporcione sea muy cercano al valor esperado. Esto se logra sumando el vector $\mathbf{p}$ a la matriz de pesos $\mathbf{W}$ y de esta forma, después de un número finito de pasos (número de veces que el vector $\mathbf{p}$ es presentado a la red), el vector $\mathbf{W}$ se aproxime asintóticamente al vector $\mathbf{p}$.

El proceso de aprendizaje puede definirse en tres reglas las cuales cubren las combinaciones de las salidas de la red, con sus correspondientes valores objetivos. Éstas reglas, junto con la función de transferencia *hardlim*, expresan lo siguiente:

$$\text{Si } t = 1 \text{ y } a = 0 \text{ entonces } \mathbf{W}^{\text{nuevo}} = \mathbf{W}^{\text{anterior}} + \mathbf{p} \tag{3.18}$$
$$\text{Si } t = 0 \text{ y } a = 1 \text{ entonces } \mathbf{W}^{\text{nuevo}} = \mathbf{W}^{\text{anterior}} - \mathbf{p} \tag{3.19}$$
$$\text{Si } t = a \text{ entonces } \mathbf{W}^{\text{nuevo}} = \mathbf{W}^{\text{anterior}} \tag{3.20}$$

Las tres condiciones anteriores pueden ser escritas en forma más compacta y generalizarse para la utilización de las funciones de transferencia *hardlims*, esto se puede realizar introduciendo el ***error*** en las reglas del aprendizaje del perceptrón, el cual se define como:

$$e = t - a \tag{3.21}$$

Así se deduce que

$$\text{Si } e = 1 \text{ entonces } \mathbf{W}^{\text{nuevo}} = \mathbf{W}^{\text{anterior}} + \mathbf{p}$$
$$\text{Si } e = -1 \text{ entonces } \mathbf{W}^{\text{nuevo}} = \mathbf{W}^{\text{anterior}} - \mathbf{p}$$
$$\text{Si } e = 0 \text{ entonces } \mathbf{W}^{\text{nuevo}} = \mathbf{W}^{\text{anterior}}$$

Lo anterior, (3.18), (3.19) y (3.20) puede ser expresado mediante una sola ecuación,

$$\mathbf{W}^{\text{nuevo}} = \mathbf{W}^{\text{anterior}} + e\mathbf{p}. \tag{3.22}$$

Aplicando el mismo razonamiento, se tiene que las ganancias están dadas mediante:

$$b^{\text{nueva}} = b^{\text{anterior}} + e. \tag{3.23}$$

La regla anterior puede ser generalizada para el caso de que se tenga múltiples perceptrones (como el modelo que se observa en la Figura 3.7(c)), sin embargo, el error estará formado por un vector $\mathbf{e}$ definido por $\mathbf{e} = \mathbf{t} - \mathbf{a}$. La *actualización* de la i-ésima fila de la matriz de peso $\mathbf{W}$ está dada por:

$${}_i\mathbf{w}^{\text{nuevo}} = {}_i\mathbf{w}^{\text{anterior}} + \mathbf{e}_i\mathbf{p}$$

mientras que el i-ésimo elemento del vector de ganancia queda como

$$b_i^{\text{nuevo}} = b_i^{\text{anterior}} + \mathbf{e}_i$$

La prueba de convergencia para este algoritmo se puede consultar en [8].

Como se mencionó, el modelo neuronal del perceptrón no puede resolver ciertos problemas en donde el conjunto de datos a resolver no sean conjuntos linealmente separables. De aquí que surgen los perceptrones multicapa y el algoritmo *backpropagation* y algunos otros algoritmos derivados de éste, los cuales pueden resolver cualquier problema ya sea o no linealmente separables.

3.4. Perceptrón multicapa

Las redes neuronales multicapa de perceptrones (MPL, *Multi-Layer Perceptron*), en términos generales son flexibles y son modelos no lineales el cual consiste de unidades (perceptones) organizadas en múltiples capas. La complejidad de los MPL puede ser cambiada al variar el número de capas y el número de unidades en cada capa (Véase [1]).

La regla de aprendizaje del perceptrón propuesto por Rosenblatt y el algoritmo LMS (*Least Mean Square*) de Widrow y Hoff (Véase [8]) fueron diseñados para el entrenamiento de redes de una sola capa. Rosenblatt y Widrow fueron consientes de las posibles limitaciones que podrían tener sus modelos, de aquí que propusieron las redes multicapa que es una generalización de la arquitectura del perceptrón simple. Sin embargo, no fueron capaces de extender sus algoritmos respectivos para esta clase de redes.

La primera descripción de un algoritmo de entrenamiento para redes multicapa (de manera general) fue propuesto por Paul Werbos en 1974. Sin embargo este algoritmo no fue aceptado dentro de la comunidad de desarrolladores de redes neuronales por el motivo de que las redes neuronales multicapa era un caso especial para la aplicación de este algoritmo (Véase [10]).

No fue que a mediados de los 80's que el algoritmo *backpropagation* fue redescubierto de manera independiente por Rumelhart, Geoffrey Hinton y Ronald Williams; David Parker y Yann Le Cun. Este algoritmo se popularizó al ser incluido en el libro *Parallel Distributed Processing* que trajo consigo un auge en las investigaciones de las redes neuronales siendo éste aprendizaje más ampliamente usado en la actualidad.

El perceptrón básico de dos capas (función de transferencia lineal en la capa de entrada y función umbral en la capa de salida) sólo puede establecer dos regiones separadas por una frontera lineal en el espacio de patrones de entrada. Por otra parte, con tres capas de neuronas se puede formar cualquier región convexa en este espacio. Las regiones convexas se forman mediante la intersección entre las regiones formadas por cada neurona de la segunda capa. Cada uno de éstos elementos se comportan como un perceptrón simple activandose su salida para los patrones de un lado del hiperplano. En la Tabla 3.1 se muestran las diferentes regiones que un perceptrón multicapa realiza al resolver el problema de la XOR cambiando el número de capas y neuronas de cada una de estas.

Para el caso de un perceptrón de cuatro capas se forman regiones de decisión arbitrariamente complejas. El proceso de separación e clases que se lleva a cabo consiste en la partición de la región deseada en pequeños hipercubos (Véase [11]).

3.4.1. *Backpropagation*

El algoritmo *backpropagation* es un tipo de aprendizaje supervisado que emplea un ***ciclo de propagación***. Una vez que se ha aplicado un patrón a la entrada de la red, éste se propaga desde la primera capa a hacia las capas superiores a través de la red hasta generar una salida. La señal de salida es comparada con la salida deseada y se calcula una

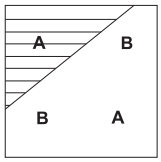
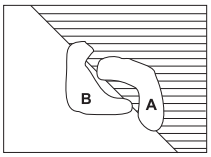
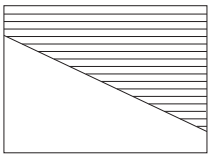
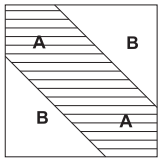
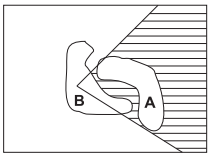
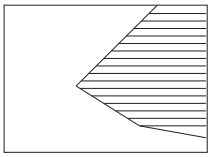
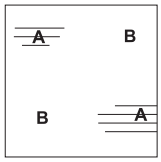
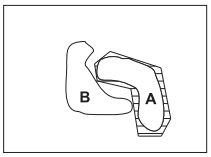
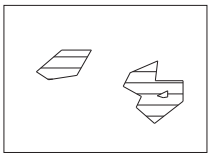
Estructura	Regiones de decisión	Problema de la XOR	Clases con regiones mezcladas	Formas de regiones más generales
2 Capas	Medio plano limitado por un hiperplano			
3 Capas	Regiones cerradas o convexas			
4 Capas	Arbitraria complejidad limitada por el número de neuronas			

Tabla 3.1: Distintas formas de las regiones generadas por un perceptrón multicapa.

señal de error para cada una de las salidas. Luego las salidas de error se propaga hacia atrás partiendo de la capa de salida hacia todas las neuronas de las capas ocultas que contribuyeron directamente a la salida.

Las neuronas de la capa oculta sólo reciben una fracción de la señal total del error, esto basándose en la contribución relativa que haya aportado cada neurona a la salida original.

Lo anterior se repite capa por capa hasta que todas las neuronas de la red hayan recibido una señal del error que describa la contribución relativa al error que haya tenido. Basándose en la señal del error percibida, se actualiza los pesos de conexión de cada neurona con el objetivo de que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento.

La importancia del desarrollo anterior consiste en que las neuronas de las capas intermedias se organizan a medida de que la red es entrenada de tal modo que las neuronas aprenden a reconocer distintas características del espacio total de entrada.

Al presentarle a la red, después del entrenamiento, un patrón de entrada arbitrario que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento. Caso contrario, las unidades de las capas ocultas tienen una tendencia a inhibir su salida si el patrón de entrada no contiene la característica para ser reconocida.

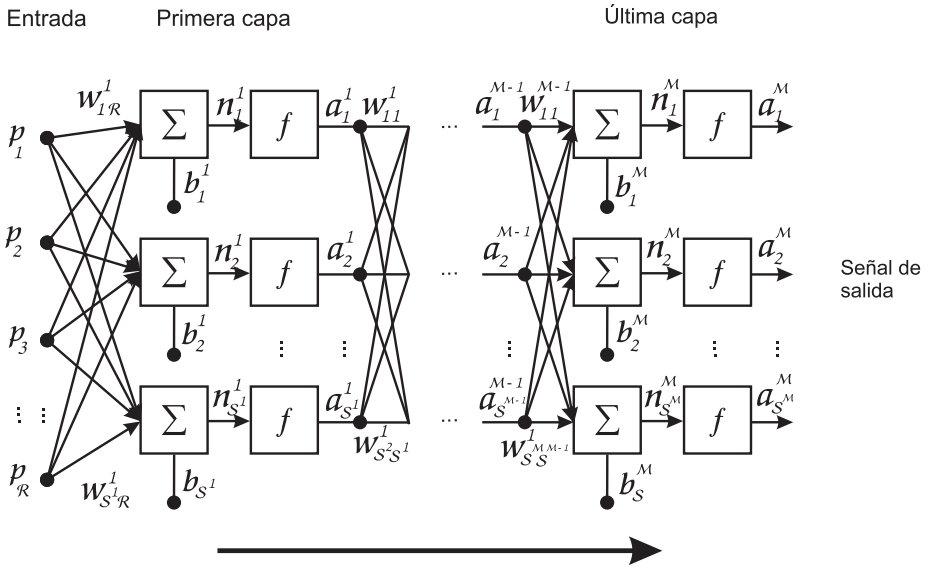


Figura 3.8: Primera fase del ciclo.

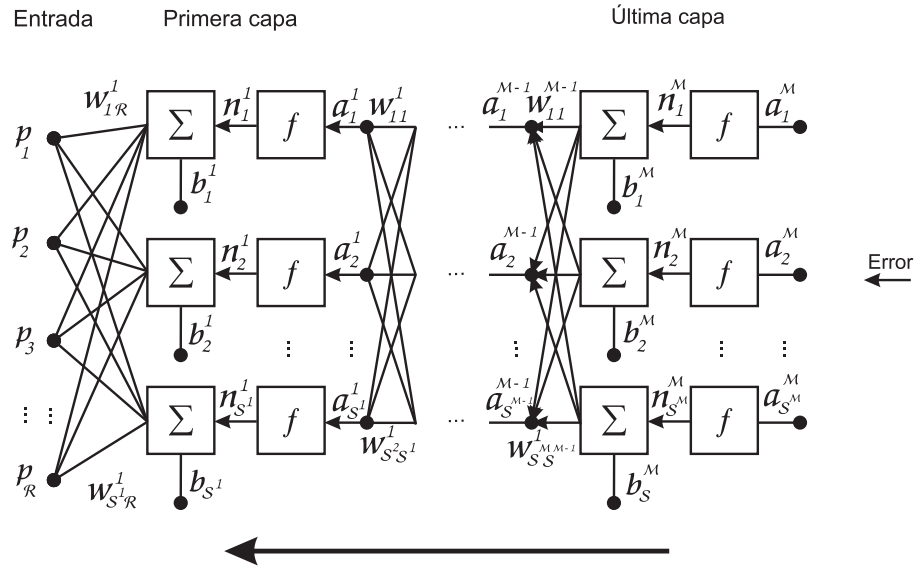


Figura 3.9: Segunda fase del ciclo.

Estructura de la red

Un MLP tiene tres características muy distintivas (Véase [10])

1. El modelo de cada neurona en la red incluye una función de transferencia no lineal el cual deberá de ser *suave*, es decir, diferenciable. La función utilizada comúnmente es

la función log-sigmoidal o tan-sigmoidal definida en (3.9) y (3.8), respectivamente.

2. La red cuenta con al menos una capa oculta de neuronas. Estas neuronas disponen a la red de un aprendizaje más completo, esto mediante la extracción de características significativas de los patrones de entrada.

3. La red exhibe un alto grado de conectividad, esto mediante las sinapsis de la red. Un cambio en la conectividad requiere un cambio en el número de conexiones o de los pesos sinapticos.

Algoritmo *Backpropagation*

Sea $R - S^1 - S^2 - \cdots - S^M$ una red con M capas. La salida de cada capa de la red pasa a convertirse en señal de entrada para la capa superior inmediata. Luego, la ecuación que describe lo anterior está dado por:

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \quad \text{para } m = 0, 1, \ldots, M-1. \tag{3.24}$$

donde $\mathbf{a}^{m+1}$ es la respectiva salida de la capa $m+1$.

Las neuronas de la primera capa reciben la señal de entrada, es decir

$$\mathbf{a}^0 = \mathbf{p}, \tag{3.25}$$

el cual provee el punto inicial de (3.24). Mientras que la señal de salida de la última capa son considerados como la señal de salida de la red:

$$\mathbf{a} = \mathbf{a}^M. \tag{3.26}$$

Como se mencionó anteriormente el algoritmo *backpropagation* es una generalización del algoritmo LMS y del algoritmo propuesto por Rosenbatt (Véase [10]), cuyo objetivo es el ajuste de los parámetros de la red (pesos) de tal manera que se minimize el ***error medio cuadrático***, esto mediante los siguientes pasos

1. Propagar la señal de entrada hacia adelante a través de la red, mediante

$$\mathbf{a}^0 = \mathbf{p} \tag{3.27}$$

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}), \text{ para } m = 0, 1, \ldots, M-1 \tag{3.28}$$

$$\mathbf{a} = \mathbf{a}^M \tag{3.29}$$

2. Propagar la *sensibilidad* hacia atrás a través de la red;

$$\mathbf{s}^M = -2\hat{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}) \tag{3.30}$$

$$\mathbf{s}^m = \hat{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})'\mathbf{s}^{m+1}, \text{ para } m = M-1, \ldots, 2, 1 \tag{3.31}$$

donde

$$\hat{\mathbf{F}}^m(\mathbf{n}^m) = \begin{pmatrix} \dfrac{\partial f^m(n_1^m)}{\partial n_1^m} & 0 & \dots & 0 \\ 0 & \dfrac{\partial f^m(n_2^m)}{\partial n_2^m} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \dfrac{\partial f^m(n_{S^m}^m)}{\partial n_{S^m}^m} \end{pmatrix} \tag{3.32}$$

y $\mathbf{n}^m$ es la entrada neta de la capa m-ésima de la red, con $m = M-1, \dots, 2, 1$.

3. Se actualizan los pesos y las ganancias usando la regla de ***pasos descendientes***,

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m(\mathbf{a}^{m-1})' \tag{3.33}$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \quad . \tag{3.34}$$

De lo anterior, (3.30) y (3.31) propagan la señal del error a través de las capas ocultas de la red de tal manera que cada neurona reciba una fracción de la señal total del error.

El factor α que se encuentra en (3.33) y (3.34) es llamado ***factor de aprendizaje*** o ***razón de aprendizaje*** cuya función es la de acelerar la velocidad de convergencia del algoritmo. Sin embargo, para valores de α muy grande los cambios en los pesos serán significativamente grandes y además existirá el riesgo de excluir el punto mínimo puesto que se avanza muy rápidamente por la superficie del error. Caso contrario con valores pequeños de α la convergencia del algoritmo se vuelve más lenta, es decir, se deberá hacer un gran número de iteraciones para poder llegar a los pesos óptimos (Véase [5]).

3.5. Variaciones del algoritmo *backpropagation*

Existen variantes del algoritmo *backpropagation* los cuales se dividen en dos categorías, esto se ilustra en el Cuadro 1. En la primera involucra el desarrollo de técnicas heurísticas que surgen de un estudio del desempeño que tiene el algoritmo *backpropagation*. Este tipo de técnicas incluyen ideas tales como *variación del factor de aprendizaje* y *momentos*.

Por otra parte, la segunda categoría esta formada por técnicas de optimización numérica, los cuales son variaciones del método de Newton tales como el algoritmo del *gradiente conjugado* y el algoritmo de *Levenberg-Marquardt*. A continuación se describirá de manera general cada una de estas variantes.

$$
\textit{Backpropagation}\begin{cases}\text{Heurísticos}\begin{cases}\text{Variación del factor de aprendizaje.}\\ \text{Momentos.}\end{cases}\\ \text{Optimización numérica}\begin{cases}\text{Gradiente conjugado}\\ \text{Levenberg-Marquardt}\end{cases}\end{cases}
$$

Cuadro 1: Variaciones del algoritmo *backpropagation.*

3.5.1. Algoritmos heurísticos

Momento

De manera general este primer método suaviza las oscilaciones que se tienen durante las trayectorias en busca del error mínimo, esto mediante la aplicación de *filtros*, tal como

$$y(k) = \gamma y(k-1) + (1-\gamma)w(k) \tag{3.35}$$

donde $w(k)$ es la entrada del filtro, $y(k)$ es la salida del filtro y γ es el ***coeficiente de momento*** con $0 \leq \gamma < 1$.

Al incrementar el valor de γ, las oscilaciones de la salida del filtro se reduce, además la media del filtro de salida es la misma que la media del filtro de entrada, aunque al incrementar γ, el filtro de salida tiene una respuesta muy lenta.

Se pueden reescribir (3.33) y (3.34) como:

$$\Delta \mathbf{W}^m(k) = -\alpha \mathbf{s}^m (\mathbf{a}^{m-1})', \tag{3.36}$$

$$\Delta \mathbf{b}^m(k) = -\alpha \mathbf{s}^m. \tag{3.37}$$

Así, al agregar el filtro de *momento* los parámetros son cambiados de la siguiente manera:

$$\Delta \mathbf{W}^m(k) = \gamma \Delta \mathbf{W}^m(k-1) - (1-\gamma)\alpha \mathbf{s}^m (\mathbf{a}^{m-1})', \tag{3.38}$$

$$\Delta \mathbf{b}^m(k) = \gamma \Delta \mathbf{b}^m(k-1) - (1-\gamma)\alpha \mathbf{s}^m \tag{3.39}$$

Con esta modificación, se tiene que la convergencia del algoritmo se incrementa sólo cuando la trayectoria en donde se está desplazando sea una dirección consistente, además de poder utilizar un factor de aprendizaje mucho mayor (Véase [10]).

Variación del factor de aprendizaje o taza de aprendizaje variable

Como se mencionó anteriormente, el valor de α determina lo que será la convergencia del algoritmo, sin embargo este tiene sus problemas al momento de elegir dicho factor. Por

cada iteración que se realice, es recomendable que el valor del factor α incremente a medida que disminuya el error de la red durante la fase del entrenamiento. Esto podría garantizar una rápida convergencia teniendo cuidado de no tomar valores que sean demasiado grande.

Existe varios diferentes enfoques de la variación del factor de aprendizaje. Aquí se presenta un enfoque en donde la *taza de aprendizaje* varia de acuerdo al rendimiento del algoritmo. Las reglas para el algoritmo *backpropagation* con taza de aprendizaje variable esta dado por:

1. Si el error cuadrático incrementa en un porcentaje ζ típicamente entre 1 % y 5 % después de la actualización de los pesos, entonces dicha actualización se descarta y el factor de aprendizaje es multiplicado por un factor $0 < \rho < 1$ y el coeficiente de momento γ, si es utilizado, se fija en cero.

2. Si el error cuadrático decrece después de actualizar los pesos, entonces dicha actualización es aceptada y el factor de aprendizaje es multiplicado por algún factor $\eta > 1$. Si γ ha sido previamente fijado a cero, éste vuelve a tomar su valor original.

3. Si el error cuadrático incrementa en menos que ζ entonces los pesos actualizados son aceptados pero el factor de aprendizaje cambia. Si γ ha sido previamente fijado a cero, éste vuelve a tomar su valor original.

Éstos algoritmos son los dos enfoques heurísticos más utilizados para modificar el algoritmo *backpropagation*. Estas modificaciones garantiza una rápida convergencia para algunos problemas sin embargo presentan dos problemas principales:

1. Requiere de un gran número de parámetros (ζ, ρ, γ), los cuales son definidos por un método de ensayo y error.

2. El algoritmo puede no converger.

3.5.2. Técnicas de optimización numérica

Gradiente conjugado

Este algoritmo no involucra el cálculo de las segundas derivadas de las variables y converge al mínimo de la función cuadrática en un número finito de iteraciones. El algoritmo consiste en lo siguiente:

1. Seleccionar como condición inicial, la dirección de $\mathbf{p}_0$ en el sentido negativo del gradiente

$$\mathbf{p}_0 = -\mathbf{g}_0, \tag{3.40}$$

donde

$$\mathbf{g}_k = \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}. \tag{3.41}$$

2. Seleccionar el factor de aprendizaje α_k que minimice la función a lo largo de la dirección

$$x_{k+1} = x_k + \alpha_k \mathbf{p}_k \tag{3.42}$$

3. Seleccionar la siguiente dirección de acuerdo con

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k+1} \tag{3.43}$$

donde

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\Delta \mathbf{g}_{k-1}^T \mathbf{p}_{k-1}}, \quad \text{o} \quad \beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}, \quad \text{o} \quad \beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \tag{3.44}$$

4. Si el algoritmo aún no converge, regresar al paso 2.

Éste algoritmo no puede ser aplicado directamente a una red neuronal puesto que el error no es una función cuadrática. Esto afecta el algoritmo de dos maneras:

1. No es hábil para minimizar la función a lo largo de una línea.
2. El error mínimo no es alcanzado en un número finito de pasos. Es por esto que el algoritmo necesita ser inicializado después de un número determinado de iteraciones.

A pesar de las limitaciones anteriores este tipo de modificación converge en pocas iteraciones e incluso para la resolución de ciertos problemas es uno de los algoritmos más rápidos para redes multicapa (Véase [5]).

Algoritmo Levenberg-Marquardt

Este algoritmo es una variación del método de Newton, el cual permite minimizar funciones que son sumas cuadráticas de funciones no lineales. Este algoritmo tiene un excelente desempeño en el entrenamiento de redes neuronales donde el rendimiento está determinado por el error cuadrático medio.

El método de Newton para la optimización del rendimiento de $F(x)$ esta dado por

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k, \tag{3.45}$$

donde $\mathbf{A}_k \equiv \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$ y $\mathbf{g}_k \equiv \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$.

Si se supone que $F(\mathbf{x})$ es una suma de funciones cuadráticas

$$F(\mathbf{x}) = \sum_{i=1}^{N} v_i^2(\mathbf{x}) = \mathbf{v}'(\mathbf{x})\mathbf{v}(\mathbf{x}), \tag{3.46}$$

entonces el j-ésimo elemento del gradiente deberá ser

$$[\nabla F(\mathbf{x})]_j = \frac{\partial F(\mathbf{x})}{\partial x_j} = 2\sum_{i=1}^{N} v_i(\mathbf{x})\frac{\partial v_i(\mathbf{x})}{\partial x_j} \tag{3.47}$$

Así, el gradiente puede ser reescrito en forma matricial

$$\nabla F(\mathbf{x}) = 2\mathbf{J}'(\mathbf{x})\mathbf{v}(\mathbf{x}), \tag{3.48}$$

donde

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{pmatrix} \tag{3.49}$$

es la ***matriz Jacobiana***.

El elemento k, j de la matriz Hessiana está expresado como:

$$[\nabla^2 F(\mathbf{x})]_{k,j} = \frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_j} = 2\sum_{i=1}^{N}\left\{\frac{\partial v_i(\mathbf{x})}{\partial x_k}\frac{\partial v_i(\mathbf{x})}{\partial x_j} + v_i(\mathbf{x})\frac{\partial^2 v_i(\mathbf{x})}{\partial x_k \partial x_j}\right\} \tag{3.50}$$

Luego la Hessiana en forma matricial queda expresado como:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}'(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}) \tag{3.51}$$

donde

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^{N} v_i(\mathbf{x})\nabla^2 v_i(\mathbf{x}) \tag{3.52}$$

Si se asume que $\mathbf{S}(\mathbf{x})$ es pequeño entonces se puede aproximar la matriz Hessiana mediante

$$\nabla^2 F(\mathbf{x}) \equiv 2\mathbf{J}'(\mathbf{x})\mathbf{J}(\mathbf{x}). \tag{3.53}$$

Así, a partir del método de Newton, se obtiene el algoritmo de Levenberg Marquardt

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\mathbf{J}'(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}\right)^{-1}\mathbf{J}'(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \tag{3.54}$$

el cual puede ser reescrito en término de incrementos

$$\Delta \mathbf{x}_k = -\Big(\mathbf{J}'(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}\Big)^{-1} \mathbf{J}'(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \tag{3.55}$$

La constante μ_k determina la tendencia del algoritmo. Cuando μ_k se incrementa dicho algoritmo se aproxima al algoritmo de pasos descendientes para factor de aprendizaje pequeño; mientras que al disminuir el valor de μ_k el algoritmo se convierte en el método de Gauss-Newton.

El algoritmo comienza con un valor pequeño para μ_k, por lo general el valor de 0.01. Si no se alcanza el valor para $F(\mathbf{x})$, entonces el paso es repetido con μ_k multiplicado por un factor $\vartheta > 1$. Por el contrario, si se produce un pequeño valor para $F(\mathbf{x})$ entonces el algoritmo es el método de Gauss-Newton, el cual hace la suposición de una rápida convergencia.

El ***índice de rendimiento*** para el entrenamiento de redes multicapa es el error medio cuadrático, es decir

$$F(\mathbf{x}) = \mathrm{E}[\mathbf{e}'\mathbf{e}] = \mathrm{E}[(\mathbf{t}-\mathbf{a})'(\mathbf{t}-\mathbf{a})] \tag{3.56}$$

Si se hace la suposición de que la ocurrencia de cada objetivo tiene la misma probabilidad entonces el error medio cuadrático es proporcional a la suma de los cuadrados de los errores sobre el objetivo Q en el conjunto de entrenamiento, es decir

$$\begin{aligned} F(\mathbf{x}) &= \sum_{q=1}^{Q} (\mathbf{t}_q - \mathbf{a}_q)'(\mathbf{t}_q - \mathbf{a}_q) \\ &= \sum_{q=1}^{Q} \mathbf{e}_q'\mathbf{e}_q = \sum_{q=1}^{Q}\sum_{j=1}^{S^M} (e_{j,q})^2 \\ &= \sum_{i=1}^{N} (v_i)^2 \end{aligned} \tag{3.57}$$

donde $e_{j,q}$ es el j-ésimo elemento del error de la q-ésimo par entrada/objetivo y se observa además que (3.46) es equivalente al índice de rendimiento (3.57).

El paso principal del algoritmo de Levenberg Marquardt es el cálculo de la matriz Jacobiana. Sin embargo, en el algoritmo *backpropagation* se calcula las derivadas de los errores al cuadrado con respecto a los pesos y ganancias de la red. Así, para el calculo de la matriz Jacobiana se realiza el calculo de las derivadas de los errores en lugar de las derivadas de los errores al cuadrado.

De lo anterior, sea el vector de error $\mathbf{v}'$ definido como

$$\begin{aligned} \mathbf{v} &= \begin{pmatrix} v_1 & v_2 & \dots & v_N \end{pmatrix} \\ &= \begin{pmatrix} e_{1,1} & e_{2,1} & \dots & e_{S^M,1} & e_{1,2} & \dots & e_{S^M,Q} \end{pmatrix} \end{aligned} \tag{3.58}$$

y el vector de parámetros

$$\begin{aligned}\mathbf{x}' &= \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} \\ &= \begin{pmatrix} w^1_{1,1} & w^1_{1,2} \dots & w^1_{S^1,R} & b^1_1 & \dots & b^1_{S^1} & w^2_{1,1} \dots & b^M_{S^M} \end{pmatrix}\end{aligned} \tag{3.59}$$

donde $N = Q \times S^M$ y $n = S^1(R+1) + S^2(S^1+1) + \cdots + S^M(S^{M-1}+1)$.

Por tanto, al realizar la sustitución en (3.49) la correspondiente matriz Jacobiana para el entrenamiento de una red multicapa puede ser reescrito como

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \frac{\partial e_{1,1}}{\partial w^1_{1,1}} & \frac{\partial e_{1,1}}{\partial w^1_{1,2}} & \cdots & \frac{\partial e_{1,1}}{\partial w^1_{S^1,R}} & \frac{\partial e_{1,1}}{\partial b^1_1} & \cdots \\ \frac{\partial e_{2,1}}{\partial w^1_{1,1}} & \frac{\partial e_{2,1}}{\partial w^1_{1,2}} & \cdots & \frac{\partial e_{2,1}}{\partial w^1_{S^1,R}} & \frac{\partial e_{2,1}}{\partial b^1_1} & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \\ \frac{\partial e_{S^M,1}}{\partial w^1_{1,1}} & \frac{\partial e_{S^M,1}}{\partial w^1_{1,2}} & \cdots & \frac{\partial e_{S^M,1}}{\partial w^1_{S^1,R}} & \frac{\partial e_{S^M,1}}{\partial b^1_1} & \cdots \\ \frac{\partial e_{1,2}}{\partial w^1_{1,1}} & \frac{\partial e_{1,2}}{\partial w^1_{1,2}} & \cdots & \frac{\partial e_{1,2}}{\partial w^1_{S^1,R}} & \frac{\partial e_{1,2}}{\partial b^1_1} & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \end{pmatrix} \tag{3.60}$$

Observe que cada elemento del Jacobiano, puede ser calculado de la siguiente manera

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial x_l} \tag{3.61}$$

luego haciendo uso de la regla cadena se tiene que

$$\frac{\partial \hat{F}}{\partial w^m_{i,j}} = \frac{\partial \hat{F}}{\partial n^m_i} \times \frac{\partial n^m_i}{\partial w^m_{i,j}} \tag{3.62}$$

donde el primer término del lado derecho corresponde a la sensibilidad de la red, es decir

$$s^m_i \equiv \frac{\partial \hat{F}}{\partial n^m_i} \tag{3.63}$$

Ahora, se define la ***sensibilidad de Marquardt*** mediante

$$\tilde{s}_{t,h}^{m} \equiv \frac{\partial v_h}{\partial n_{i,q}^{m}} = \frac{\partial e_{k,q}}{\partial n_{i,q}^{m}}, \tag{3.64}$$

donde $h = (q-1)S^M + k$.

Así, los elementos del Jacobiano puede ser calculado mediante

$$\begin{aligned}
[\mathbf{J}]_{h,i} &= \frac{\partial v_h}{\partial x_i} = \frac{\partial e_{k,q}}{\partial w_{i,j}^{m}} \\
&= \frac{\partial e_{k,q}}{\partial n_{i,q}^{m}} \times \frac{\partial n_{i,q}^{m}}{\partial w_{i,j}^{m}} = \tilde{s}_{i,h} \times a_{j,q}^{m-1} \\
&= \tilde{s}_{i,h}^{m} \times a_{j,q}^{m-1},
\end{aligned} \tag{3.65}$$

o cuando x_i sea una ganancia,

$$\begin{aligned}
[\mathbf{J}]_{h,i} &= \frac{\partial v_h}{\partial x_i} = \frac{\partial e_{k,q}}{\partial b_{i}^{m}} \\
&= \frac{\partial e_{k,q}}{\partial n_{i,q}^{m}} \times \frac{\partial n_{i,q}^{m}}{\partial b_{i}^{m}} = \tilde{s}_{i,h} \times \frac{\partial n_{i,q}^{m}}{\partial b_{i}^{m}} \\
&= \tilde{s}_{i,h}^{m}.
\end{aligned} \tag{3.66}$$

Para la sensibilidad Marquardt de la última capa se tiene

$$\begin{aligned}
\tilde{s}_{i,h}^{M} &= \frac{\partial v_h}{\partial n_{i,q}^{M}} = \frac{\partial e_{k,q}}{\partial n_{i,q}^{M}} \\
&= \frac{\partial (t_{k,q} - a_{k,q}^{M})}{\partial n_{i,q}^{M}} = -\frac{\partial a_{k,q}^{M}}{\partial n_{i,q}^{M}} \\
&= \begin{cases} -f^{M}(n_{i,q}^{M}), & \text{para } \; i = k; \\ \\ 0, & \text{para } \; i \neq k. \end{cases}
\end{aligned} \tag{3.67}$$

Por tanto, cuando la entrada $\mathbf{p}_q$ haya sido aplicado a la red y la correspondiente salida $\mathbf{a}_q^M$ haya sido calculada, el algoritmo *backpropagation* Levenberg Marquardt es inicializado con

$$\tilde{\mathbf{S}}_q = -\hat{F}^{M}(\mathbf{n}_q^{M}), \tag{3.68}$$

donde $\hat{\mathbf{F}}^{M}(\mathbf{n}^{M})$ esta definido mediante (3.32). Cada columna de la matriz $\tilde{\mathbf{S}}_q^M$ deberá ser la retropropagación a través de la red usando (3.31) al producir una fila de la matriz Jacobiano. Además las columnas pueden ser propagadas conjuntamente mediante

$$\tilde{\mathbf{S}}_q^m = \tilde{\mathbf{F}}^m(\mathbf{n}_q^m)(\mathbf{W}^{m+1})'\tilde{\mathbf{S}_q^{m+1}}. \tag{3.69}$$

Así, la sensibilidad total Marquardt para cada capa puede entonces ser creada por medio de la matriz aumentada calculada por cada entrada:

$$\tilde{\mathbf{S}}^m = \left[\tilde{\mathbf{S}}_1^m|\tilde{\mathbf{S}}_2^m|\cdots|\tilde{\mathbf{S}}_Q^m\right] \tag{3.70}$$

Para cada nueva entrada que es presentada a la red, los vectores de sensibilidad son propagados hacia atrás, esto debido a que se ha calculado cada error en forma individual en lugar de derivar la suma al cuadrado de los errores. Luego, para cada entrada aplicada a la red habrá S^M errores, uno por cada elemento de salida de la red y por cada error generará una fila de la matriz Jacobiana.

Las iteraciones del algoritmo *backpropagation* Levenberg Marquardt se resumen en los siguientes pasos

1. Presentar todas las entradas a la red y calcular las correspondientes salidas de la red (por medio de (3.27) y (3.28)), además de los errores $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q^M$. Calcular la suma de los cuadrados de todas las entradas, $F(\mathbf{x})$, ésto último mediante (3.57).

2. Calcular la matriz Jacobiana (3.60) además de las sensibilidades con la relación de recurrencia (3.69), después inicializar con (3.68). Calcular (3.70) y calcular los elementos de la matriz Jacobiana mediante (3.65) y (3.66).

3. Resolver (3.55) para obtener Δx_k.

4. Recalcular la suma de los cuadrados de los errores usando $x_k + \Delta x_k$. Si ésta suma es pequeña en comparación con la calculada en el paso 1, entonces dividir μ por ζ y se toma $x_{k+1} = x_k + \Delta x_k$ y regresar al paso 1. Si la suma no se reduce multiplicar μ por ζ y regresar al paso 3.

En este trabajo se centrará en utilizar éste último algoritmo puesto que presenta una convergencia mucho más rápida que los demás algoritmos presentados anteriormente. En el siguiente capítulo se realiza los correspondientes análisis de los datos que se disponen y posteriormente el pronóstico utilizando los enfoques previamente revisados.

Capítulo 4
Aplicación

La electricidad constituye una de las principales fuentes energéticas con las que cuenta la humanidad. Su empleo abarca un amplio abanico de actividades que se extiende desde los usos exclusivamente industriales, hasta el consumo doméstico de las familias.

En el mercado eléctrico se ha tenido la preocupación de conocer con exactitud el consumo previsible en un tiempo relativamente corto, esto con el fin de ofertar precios que se adapten en lo posible a éstas demandas y poder planificar las producciones de los centros de distribución. Por tanto, el objetivo principal de los mercados es mejorar la calidad de los servicios y esto depende de la capacidad que tiene la empresa de proveer y distribuir este servicio.

Durante las décadas de los 50's y 60's las técnicas más usadas para realizar predicción eran las *técnicas de predicción simples* tales como la extrapolación. Esto debido a la estabilidad de los precios y de que la tendencia demográfica era predecible en muchas áreas geográficas. Sin embargo, dicha predicción no ha sido lo suficientemente buena en la actualidad y se han realizado investigaciones para poder realizar predicciones aceptables. Tal es el caso de los métodos estadísticos y el enfoque de las redes neuronales artificiales, los cuales se han estudiado en los capítulos anteriores.

4.1. Datos

Para realizar este trabajo se dispone de un conjunto de datos que corresponden al registro de demanda máxima mensual de energía eléctrica en una subestación de un sistema de distribución de la Comisión Federal de Electricidad, cuyos valores están expresados en megawatts (MW). Estos registros corresponden al periodo de enero de 1994 a diciembre de 2006. El análisis se realizó eligiendo uno de los dos transformadores que componen la subestación, con el objetivo de realizar la comparación de la predicción (proceso de simulación) se tomaron los datos hasta junio de 2006 dejando los últimos seis meses (julio a diciembre de 2006) para tal fin. Resultados previos fueron expuestos en [13]. En la Figura 4.1 se ilustra la serie de estos datos.

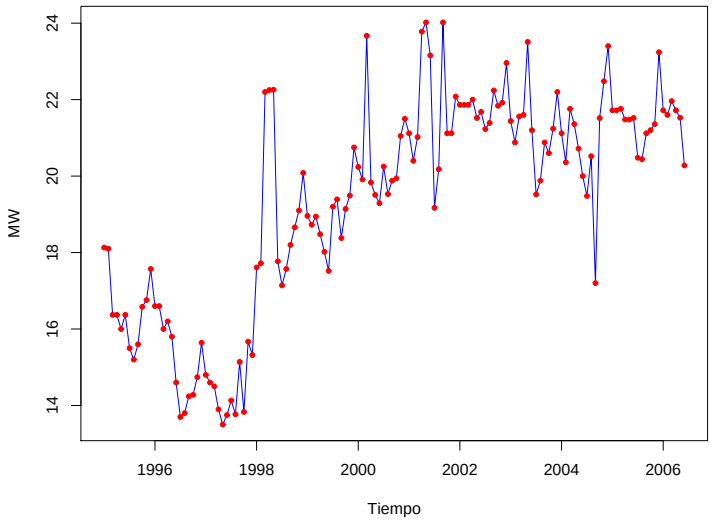


Figura 4.1: Demanda máxima mensual. Enero 1994 - junio 2006.

4.2. Pronóstico con series de tiempo

Para poder realizar pronósticos utilizando este tipo de metodología comúnmente se realizan los siguientes pasos (Véase [15]).

- Ajuste de modelo.
- Validación del modelo.
- Pronóstico.

Para el ajuste del modelo, primero se tiene que tener una serie estacionaria, y como se observa en la Figura 4.1 la tendencia es no constante por lo que se tiene una serie no estacionaria. Esto se corrobora en los correlogramas de la serie presentados en la Figura 4.2 en donde los valores de la ACF decrecen o decaen muy lentamente.

Como se mostró en la sección 2.4 página 17, una de las técnicas para obtener una serie estacionaria es aplicar sucesivamente diferencias, cuyo resultado será una serie con componentes de tendencia y estacionalidad ausentes siendo además estacionaria. En la Figura 4.3 se muestra la serie que resulto de aplicar la primera diferencia y como se observa tiene una tendencia constante. En la Figura 4.4 se muestra los correlogramas de ésta, con lo que se concluye la estacionalidad de la serie.

Una vez que se obtuvo la serie estacionaria se propusieron algunos modelos que ajusten a la serie, observando los correlogramas de la serie resultante (Véase Figura 4.4). Dichos

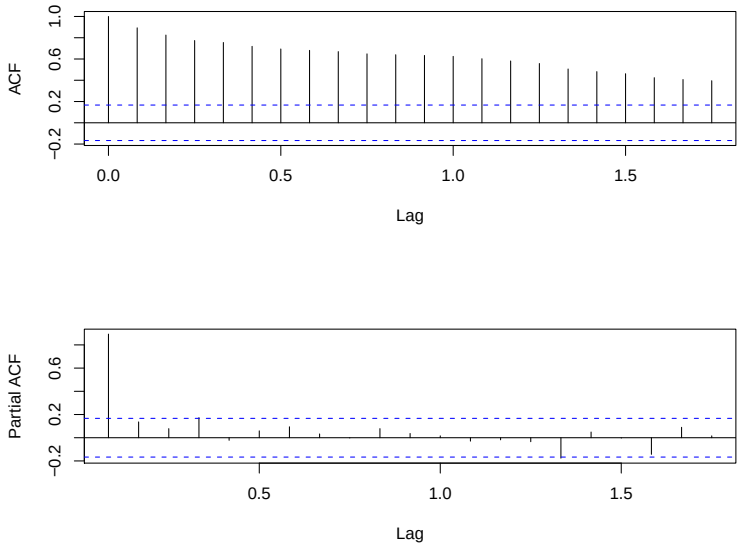


Figura 4.2: Correlogramas.

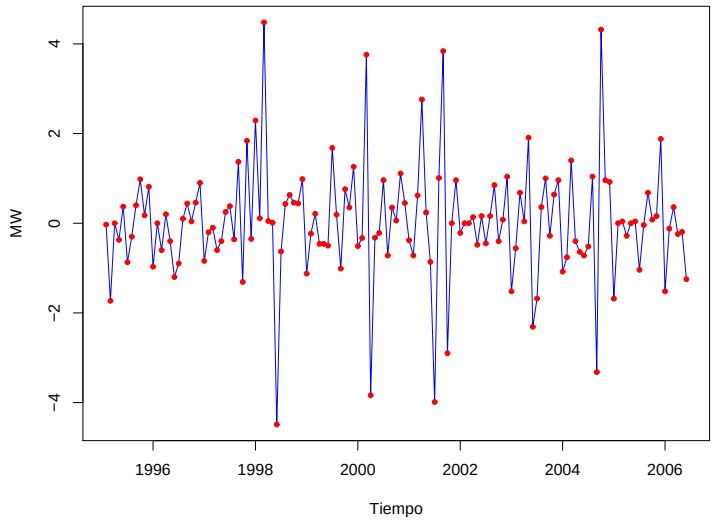


Figura 4.3: Serie resultante de aplicar una diferencia.

modelos se muestran en la Tabla 4.1, en la cual se presenta la varianza estimada, la log-verosimilitud y el valor del estadístico AICC.

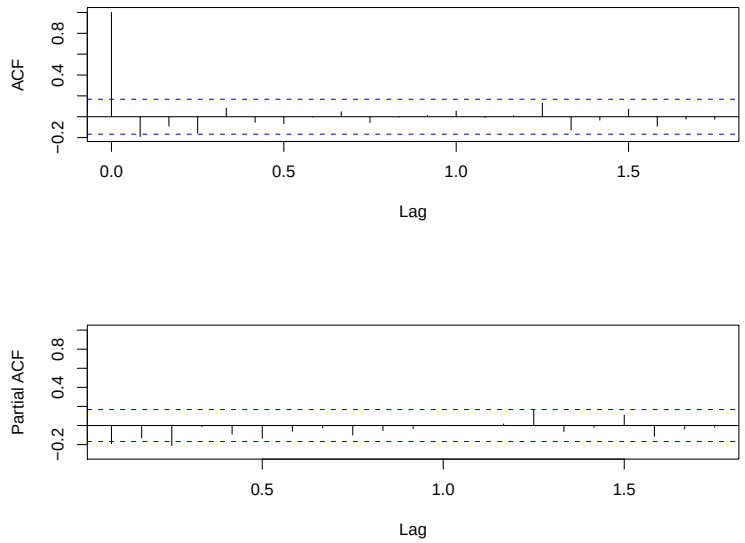


Figura 4.4: Correlogramas de la serie diferenciada.

Modelo	Varianza Estimada	Log-Verosimilitud	Criterio AICC
$ARIMA(3,1,1)$	1.48	-221.21	452.88
$ARIMA(3,1,3)$	1.38	-218.52	451.91
$ARIMA(3,1,2)$	1.46	-220.71	454.07
$ARIMA(2,1,2)$	1.48	-221.21	452.87
$ARIMA(0,1,2)$	1.52	-223.24	452.65
$ARIMA(0,1,1)$	1.58	-225.69	455.46
$ARIMA(2,1,1)$	1.48	-221.44	451.19
$ARIMA(1,1,1)$	1.48	-221.48	449.13
$ARIMA(1,1,2)$	1.48	-221.45	451.20
$ARIMA(1,1,0)$	1.61	-226.83	457.76

Tabla 4.1: Modelos propuestos.

Como se mostró en la sección 2.8.1 inciso 2 de la página 40, el criterio para elegir uno de los modelos propuestos es el criterio AICC, el cual es elegir aquel cuyo valor sea mínimo. De aquí que el modelo que mejor se ajusto a los datos es el $ARIMA(1,1,1)$.

Los valores estimados para los parámetros de éste modelo son:

$$\hat{\phi} = 0.54 \qquad \hat{\theta} = -0.84$$

así, el modelo ARIMA$(1,1,1)$ estimado puede ser reescrito como:

$$X_t = 1.5415X_{t-1} - 0.5415X_{t-2} + Z_t + 0.8387Z_{t-1}, \tag{4.1}$$

donde $Z_t \sim \text{WN}(0, 1.48)$.

Para la validación de este modelo se realizó un análisis de los residuales utilizando la función de autocorrelación muestral de éste. De acuerdo con la sección 2.7 en la página 37 permite rechazar el modelo siempre y cuando más de dos o tres de cada 40 quedan fuera de los límites de la banda de confianza o si por lo menos uno está muy por fuera de éstos límites. En la Figura 4.5 se ilustra la ACF muestral de los residuales, y como se observa, los 40 valores caen dentro de la banda de confianza.

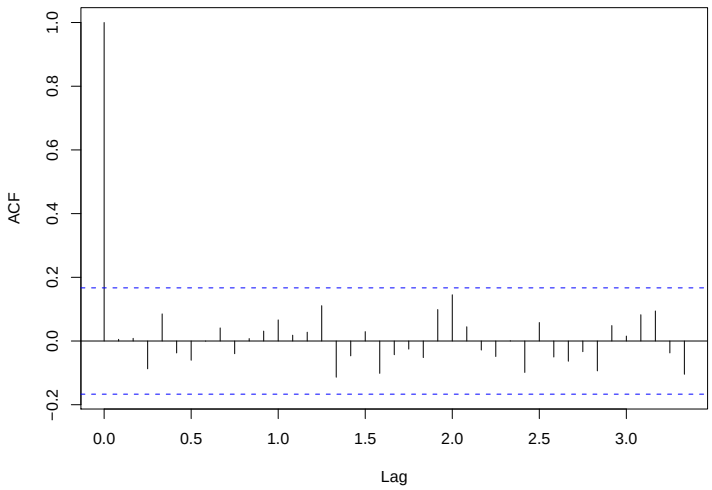


Figura 4.5: Función de autocorrelación muestral de los residuales.

Con el análisis anterior se concluye que el modelo ARIMA$(1,1,1)$ que se ilustra en la Tabla 4.1 proporciona un buen ajuste a los datos. En la Figura 4.6 se ilustra el modelo ajustado a la serie.

Finalmente, al tener el modelo se llevó a cabo el pronóstico de los últimos seis meses del año 2006. En la Tabla 4.2 se muestra el valor real, el pronostico y el error absoluto de cada periodo. Como se observa éstos aumentan conforme se predice un número de periodo mayor, pero aún así para estos 6 periodos el error es pequeño.

En la Figura 4.7 se muestra ésta predicción junto con una banda de confianza del 95 %, y en la Figura 4.8 se ilustra la comparativa de la predicción del modelo utilizado junto con los valores reales.

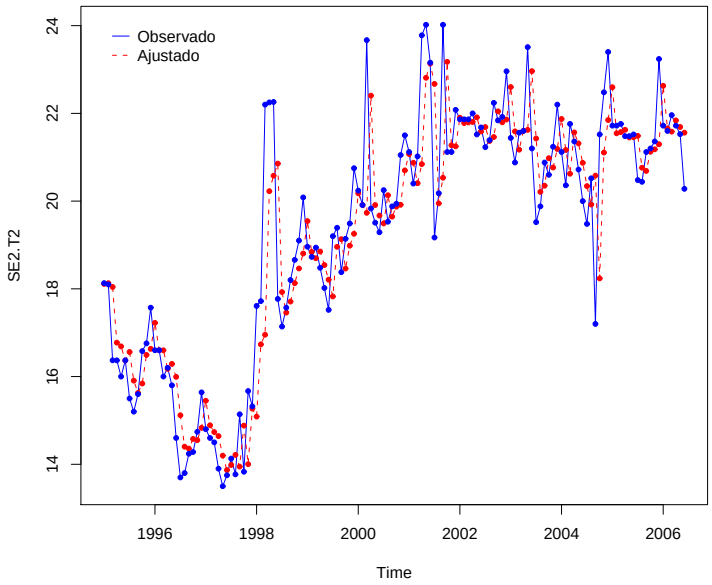


Figura 4.6: Serie original y modelo ajustado.

Periodo	Valor Real	Pronostico	Error absoluto
2006.7	20.40	20.68	0.28
2006.8	21.48	20.89	0.59
2006.9	22.32	21.01	1.31
2006.10	23.04	21.07	1.97
2006.11	22.56	21.11	1.45
2006.12	23.16	21.13	2.03

Tabla 4.2: Pronósticos usando el modelo ARIMA$(1, 1, 1)$.

4.3. Pronóstico con redes neuronales

Para tener una red neuronal la cual se ajustará de manera adecuada a los datos, se tuvieron que realizar varias configuraciones que van desde el número de capas ocultas, el número de neuronas por cada capa y las funciones de transferencias que fueron utilizadas. El algoritmo de aprendizaje que se utilizó fue la variación de *backpropagation* llamado Levenberg-Marquardt que fue estudiado en el Capítulo anterior.

Una de las características que compartieron todos los modelos desarrollados fue el diseño del conjunto de entrenamiento. El patrón de entrada consistía en un vector de dos

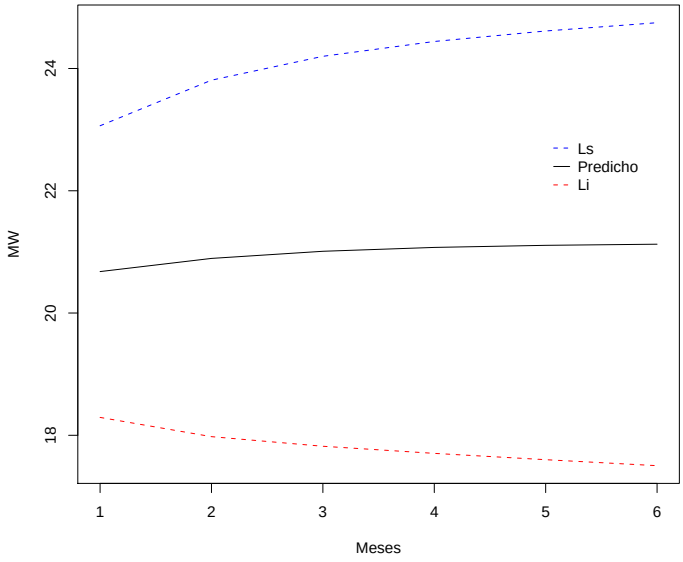


Figura 4.7: Pronóstico del modelo.

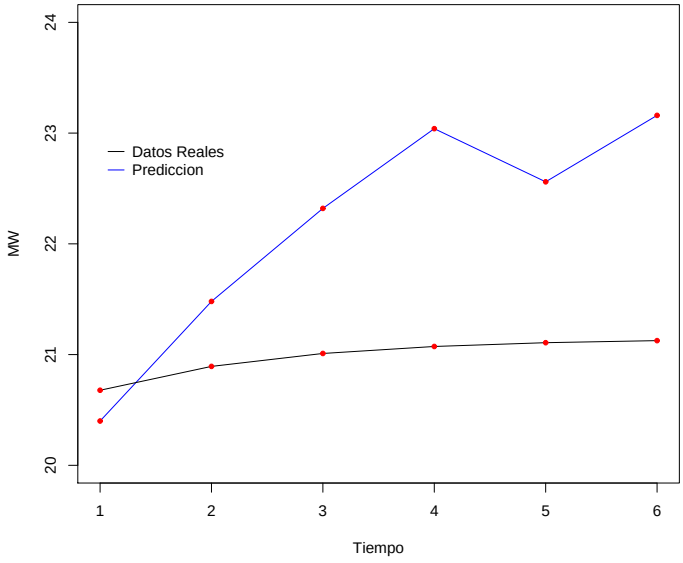


Figura 4.8: Comparativa gráfica.

componentes, en la primera correspondía al año (número entero que va desde 1994 al año 2006) y como segunda componente el mes (valor entero que va desde 1, el cual representa al mes de enero, al 12 que representa al mes de diciembre).

Los valores esperados de cada uno de los patrones de entrada fueron normalizado puesto que los valores de las salidas que se generaban en las redes estaban en el intervalo $[-1, 1]$. Dicha normalización se llevó acabo de la siguiente manera:

$$\text{Demanda} = \frac{\text{Valor de la demanda}}{\text{Demanda máxima}}.$$

La primera red que se analizó fue la red $2:2:1:1$, cuyas características se muestran en la Tabla 4.3.

Inicialización de los pesos	Funciones de transferencias	*Performance*	Número de iteraciones	Factor de aprendizaje
Aleatorio	Tangente sigmoidal Función lineal	0.003	500	0.2

Tabla 4.3: Configuración de la red $2:2:1:1$

Los resultados que se obtuvieron de esta red no fueron muy buenos debido a que la red no pudo ajustar de manera aceptable al conjunto de datos y el pronóstico que proporcionó tampoco lo era, salvo que para los mes de agosto y diciembre los datos predichos se acercaban bastante al valor real. En las Figuras 4.9, 4.10 y 4.11 se muestran el ajuste de los datos, la predicción y la evolución del *performance* de la red, respectivamente.

Aunque los valores proporcionados por la red no estaban muy próximos a los datos, se podría realizar modificaciones a la estructura de ésta para que se tuviera un buen ajuste a estos. Con lo anterior se eligió la red cuya estructura es $1:2:16:12:1$ con las matrices de pesos iniciales inicializados con ciertos valores, los cuales fueron resultados de la implementación de otras redes de la misma arquitectura salvo que sus pesos iniciales fueron aleatorios.

La configuración de la red anterior se ilustra en la Tabla 4.4. Como se muestra en la Figura 4.12, el rendimiento de la red es mucho mejor que la red anterior, además de que el ajuste a los datos es bueno como se exhibe en la Figura 4.13. En la Figura 4.14 se ilustra el pronóstico de los últimos 6 meses del año 2006 y en la Tabla 4.4 los correspondientes valores de pronóstico y los errores absolutos de cada uno de éstos.

De las Figuras 4.10 y 4.14 se observa de que ésta última tiene un mejor pronóstico los primeros cinco meses.

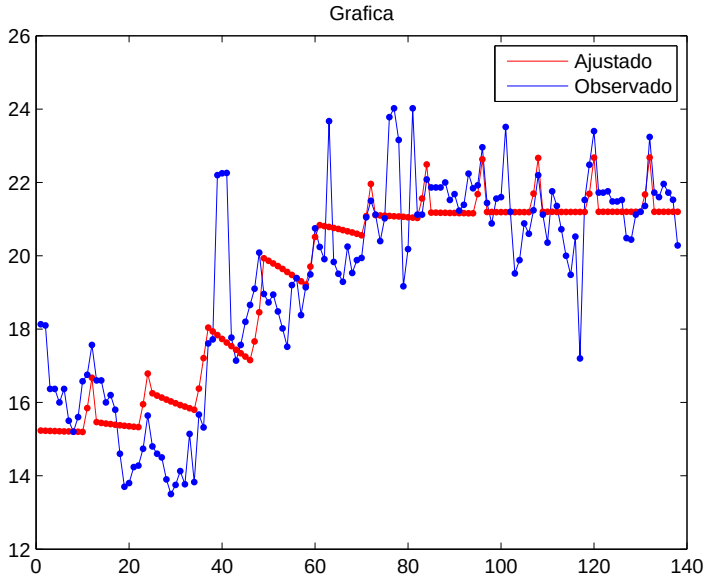


Figura 4.9: Ajuste de la red $2:2:1:1$.

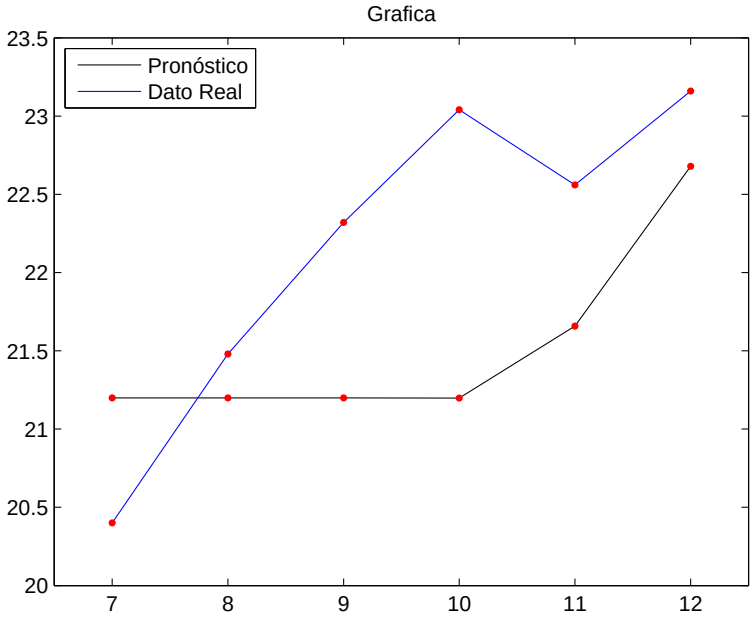


Figura 4.10: Comparación de pronóstico.

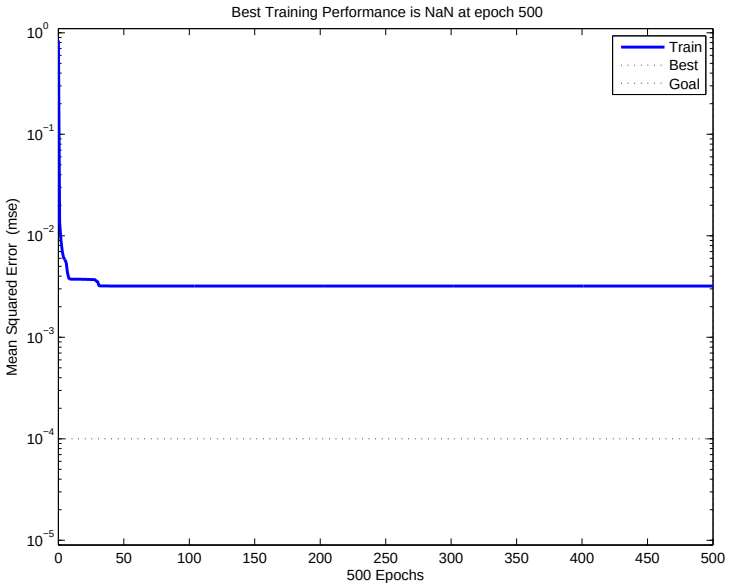


Figura 4.11: Evolución del *performance* de la red $2:2:1:1$.

Inicialización de los pesos	Funciones de transferencias	*Performance*	Número de iteraciones	Factor de aprendizaje
Configuración dada	Tangente sigmoidal Tangente sigmoidal Funcion lineal	9.91×10^{-5}	1	0.2

Tabla 4.4: Configuración de la red $1:2:16:12:1$.

Periodo	Dato Real	Pronosticado	Error absoluto
2006.7	20.40	20.21	0.19
2006.8	21.48	21.62	0.14
2006.9	22.32	22.26	0.06
2006.10	23.04	22.22	0.82
2006.11	22.56	21.60	0.96
2006.12	23.16	20.81	2.35

Tabla 4.5: Resultados obtenidos de la red $1:2:16:12:1$.

4.4. Comparación

Una vez que se han obtenido la predicción de éstos periodos mediante los dos modelos, se realiza un pequeño análisis comparativo acerca de cada uno de éstos.

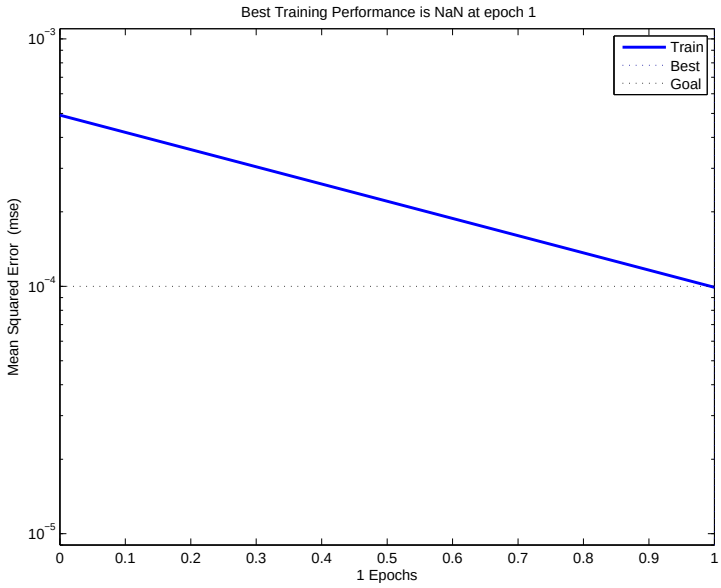


Figura 4.12: *Performance*.

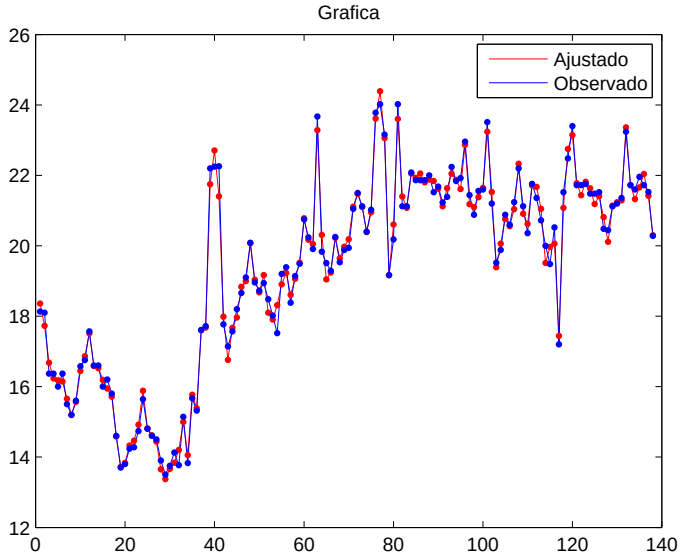


Figura 4.13: Comparativa del ajuste con los datos observados.

Para el modelo de serie de tiempo, se observó que para realizar inferencia en el conjunto de datos, la serie debería de ser estacionaria y con la técnica vista en la sección 2.8.1, la aplicación de una sola diferencia basto para que la serie resultante fuera estacionaria. Una

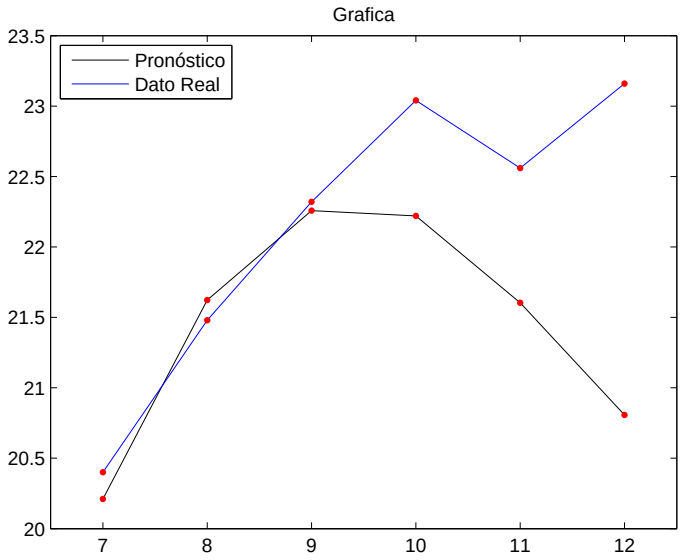


Figura 4.14: Comparación entre el pronóstico de la red contra los datos reales.

vez que se obtuvo ésta serie, se siguió un conjunto de pasos los cuales proporcionaba, en primer lugar, modelos que se ajustaran al modelo, en seguida la selección de uno de ellos mediante un criterio y finalmente se prosiguió la elaboración del pronóstico.

Por otra parte, la metodología usada para la elaboración de la red neuronal fue muy distinta. Mientras que en la metodología de series de tiempo en el seguimiento de los pasos proporciona un modelo en específico, en las redes neuronales no se tiene esto, puesto que en primer lugar no se conoce de alguna manera que topología es la adecuada para la resolución del problema. Se tuvieron de realizar diversos modelos y cada uno de ellos entrenarlos y observar si la red proporcionaba datos que se ajustase a la serie o no.

Aunque el desarrollo del modelo de la red neuronal fue a *prueba y error*, los resultados de predicción que se obtuvieron fueron mucho mejor que los proporcionados por el modelo de serie de tiempo. En la Figura 4.15 se muestra la comparativa gráfica y se observa que la curva dada por la red neuronal esta por arriba de la curva del modelo ARIMA$(1, 1, 1)$, al menos en los primeros cinco periodos.

Periodo	Dato Real	Pronostico Serie de tiempo	Pronosticado Red neuronal	Error absoluto Serie de tiempo	Error absoluto Red neuronal
2006.7	20.40	20.21	20.68	0.28	0.19
2006.8	21.48	21.62	20.89	0.59	0.14
2006.9	22.32	22.26	21.01	1.31	0.06
2006.10	23.04	22.22	21.07	1.97	0.82
2006.11	22.56	21.60	21.11	1.45	0.96
2006.12	23.16	20.81	21.13	2.03	2.35

Tabla 4.6: Resultados obtenidos.

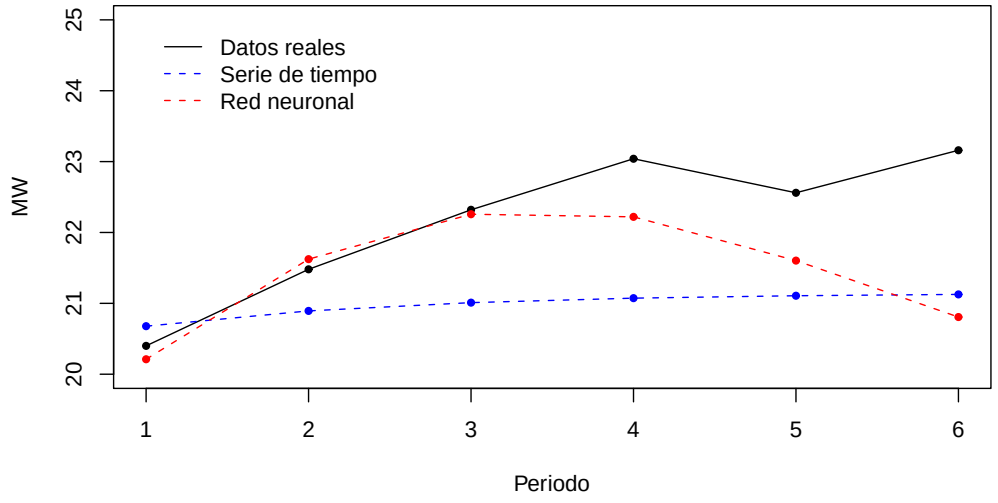


Figura 4.15: Comparación gráfica de los modelos de predicción.

Capítulo 5
Conclusiones

El objetivo de esta tesis fue el encontrar modelos de series de tiempo y de redes neuronales los cuales fueran capaz de pronosticar la demanda máxima de energía eléctrica en un periodo corto, además de realizar una comparación entre éstos.

En el modelo de serie de tiempo antes de realizar el pronóstico se aplicó diferencia a la serie para que resultara estacionaria y a partir de ésta se procedió con el pronóstico. Caso contrario para el modelo de red neuronal se normalizaron los valores esperados de cada uno de los patrones de entradas, esto debido a que los valores que proporcionaba cada red se encontraban en el intervalo $[-1, 1]$.

Las predicciones que proporcionó la red neuronal, en los cinco primeros meses, tuvieron un error menor que los valores proporcionados por el modelo de serie de tiempo (Véase Figura 4.15). Cabe destacar que conforme se va aumentando el periodo al cual se desea pronosticar, el error absoluto de la predicción usando series de tiempo incrementa, además de que existirá un periodo en donde a partir de este, los valores que la serie proporcione serán constantes.

En la red neuronal, no se conoce con exactitud el comportamiento que pueda tener la red conforme se aumenta el número de periodos que se desea pronosticar, pues como se observa en la Figura 4.14, los periodos correspondientes a los meses de octubre a diciembre, tiene un comportamiento parabólico, sin embargo este comportamiento podría no ser descrito por los demás periodos que se desean predecir.

Al desarrollar el modelo de redes neuronales no existe previamente una estructura preliminar para poder resolver el problema que se plantea, pues bien solo se afirma de la existencia de una estructura (al menos) el cual pueda dar respuestas adecuadas al problema.

Con lo anterior, al desarrollar las redes, se encontraron muchas estructuras de las cuales se ajustaban muy bien a los datos, sin embargo los errores de predicción estaban por encima de los errores dados por el modelo de serie de tiempo.

El objetivo de esta tesis fue alcanzado y se concluye que las redes neuronales proporcionan buenas predicciones comparadas con las predicciones del modelo de serie de tiempo, salvo que, como se comento anteriormente, se tuvo que buscar la estructura idónea.

Código

En este apartado se muestra el código generado para el desarrollo de la red neuronal para el ajuste y el pronóstico de los datos. Este código se ha dividido en tres partes: la primera, construye la red junto con la estructura adecuada, posteriormente la visualización del ajuste junto con la gráfica original y por último la comparativa gráfica del pronóstico junto con los datos reales.

Código 1 Construcción y entrenamiento de la red neuronal.

```
clear
clc
data=load ('Datos.txt');%Obtencion de los datos

%Construcción del vector de entrada y de los valores esperados y normalización de los TARGET
P=(data(1:138,1:2))';
T=(data(1:138,3))';
T1=T;
maximo=max(data(1:144,3));
T=T/maximo;

%Definición de la estructura de la red

net=newff([1995 2006; 1 12],[16,12,1],{'tansig','tansig','purelin'},'trainlm');
net=init(net);
net.trainparam.goal=1e-4;
net.trainparam.epochs=500;
net.trainparam.lr=0.2;

%Inicialización de los pesos

net.IW{1,1}=load('PrimeraCapaWI.txt');
net.b{1}=load('PrimeraCapabI.txt');
net.LW{2,1}=load('SegundaCapaWI.txt');
net.b{2}=load('SegundaCapabI.txt');
net.LW{3,2}=load('TerceraCapaWI.txt');
net.b{3}=load('TerceraCapabI.txt');

[net,tr]=train(net,P,T);
```

Código 2 Simulación de la red neuronal.

```
x=1:138;

predicho=[];
predicho=cat(2,predicho,sim(net,P(:,1)));
for i=2:138
    X=sim(net,P(:,i));
    predicho=cat(2,predicho,X);
end

figure(1)
predicho=predicho*maximo;
plot(1:138,predicho,'r', 1:138,T1,'b', 1:138,predicho,'r.', 1:138,T1,'b.',...
     'LineWidth',0.5, 'MarkerSize',10);
legend({'Ajustado','Observado'})
title('Grafica')
```

Código 3 Comparativa gráfica.

```
prueba=[2006 2006 2006 2006 2006 2006 2006 2006 2006 2006 2006 2006; 1 2 3 4 ...
        5 6 7 8 9 10 11 12];

dados=[20.4 21.48 22.32 23.04 22.56 23.16];

predicho1=[];
predicho1=cat(2,predicho1,sim(net,prueba(:,7)));
for i=8:12
    X=sim(net,prueba(:,i));
    predicho1=cat(2,predicho1,X);
end

predicho1=predicho1*maximo;
figure(2)
plot(7:12,predicho1,'k', 7:12,dados,'b', 7:12,predicho1,'r.', 7:12,dados,'r.',...
     'LineWidth',0.5, 'MarkerSize',10);
legend({'Pronóstico','Dato Real'},'Location','NorthWest')
title('Grafica')
axis([6.5 12.5 20 23.5])
error=dados-predicho1
```

Pesos sinápticos

En este apartado se muestran los pesos iniciales y finales que fueron utilizados y obtenidos en el aprendizaje de la red neuronal.

Primeramente se muestran las matrices o vectores iniciales de cada una de las capas ocultas de la red, los cuales se ilustran en la Matriz 1, Matriz 2, Matriz 3. Las matrices que fueron obtenidas después del proceso de aprendizaje se ilustran en la Matriz 4, Matriz 5 y Matriz 6.

$$\mathbf{W}^{11} = \begin{pmatrix} 0.84 & 0.73 \\ -0.65 & 1.34 \\ -0.22 & 0.49 \\ -1.01 & -0.04 \\ -0.85 & 1.53 \\ 0.71 & 0.58 \\ -0.91 & 0.32 \\ -0.93 & 0.54 \\ 0.93 & 0.40 \\ 0.56 & -1.38 \\ -1.01 & -0.76 \\ -1.02 & 1.22 \\ 0.74 & -0.51 \\ -1.00 & -0.34 \\ -0.78 & -0.13 \\ -0.46 & -0.31 \end{pmatrix} \qquad \mathbf{b}_1 = \begin{pmatrix} -1679.8 \\ 1293.2 \\ 469.7 \\ 2020.8 \\ 1699.9 \\ -1416.2 \\ 1827.0 \\ 1861.6 \\ -1864.5 \\ -1109.5 \\ 2032.6 \\ 2034.1 \\ -1483.9 \\ 2005.8 \\ 1560.1 \\ 919.7 \end{pmatrix}$$

Matriz 1: Pesos y ganancias correspondientes a la primera capa.

$$W^{21} = \begin{pmatrix} -0.53 & -0.07 & -0.46 & 0.20 & -0.85 & -0.04 & 0.11 & -0.63 & -0.49 & 0.25 & -0.23 & 0.42 & 0.65 & -0.88 & -0.92 & -0.47 \\ -0.19 & 0.06 & -0.54 & 0.35 & -0.90 & 1.23 & 0.35 & 0.51 & 0.15 & -0.85 & -0.17 & -0.16 & 0.17 & 0.13 & -0.22 & 0.23 \\ -0.15 & 0.49 & -0.14 & 0.63 & -0.45 & -0.40 & 0.65 & 0.32 & 0.50 & 0.21 & -0.38 & 0.09 & -0.62 & -1.13 & 1.35 & 1.02 \\ -0.88 & -0.03 & -0.00 & 1.41 & 0.73 & 0.61 & 0.87 & 0.80 & 0.25 & 0.38 & -0.13 & -0.20 & -0.21 & 1.03 & 0.08 & 0.85 \\ -0.12 & -0.22 & -0.23 & -0.10 & 0.84 & 0.01 & 0.99 & 0.51 & 0.82 & 0.10 & 0.15 & -0.70 & -0.28 & -0.67 & 0.37 & -0.87 \\ 0.08 & -0.74 & 0.54 & 0.26 & 0.28 & 1.24 & -0.20 & 0.44 & 0.07 & -0.63 & 0.53 & -0.55 & 0.83 & -0.33 & 1.16 & 0.56 \\ -0.28 & -0.31 & -0.31 & -0.00 & 0.14 & -0.85 & 0.55 & 0.19 & -0.25 & -0.53 & -0.36 & 0.59 & -0.26 & -0.04 & 0.79 & 0.88 \\ 0.92 & 0.02 & 0.81 & 0.13 & -0.49 & 0.17 & -0.15 & 0.72 & -0.39 & 0.10 & 0.46 & -0.85 & 0.97 & -1.15 & 0.25 & -0.55 \\ 0.29 & 0.09 & -0.24 & 0.34 & -0.18 & -0.32 & 0.46 & -0.66 & 0.03 & -0.95 & 1.18 & 0.39 & 0.44 & 0.07 & -0.92 & -0.90 \\ 0.96 & -0.37 & -0.49 & 0.46 & -0.05 & 1.38 & 0.21 & 0.79 & 0.09 & -1.00 & 0.30 & -0.39 & 0.88 & 0.30 & 0.69 & 0.80 \\ 0.46 & 0.57 & 0.25 & -1.44 & 0.86 & -0.65 & -0.47 & 0.20 & 0.08 & -0.35 & -0.14 & 0.86 & 0.63 & 1.46 & -0.09 & -0.46 \\ -0.03 & 0.33 & -1.02 & -0.14 & 0.79 & 1.74 & -0.23 & -0.16 & 0.02 & 0.13 & 0.09 & -0.01 & -0.13 & -0.03 & -0.77 & 0.24 \end{pmatrix}$$

$$\mathbf{b}_2 = \begin{pmatrix} 1.62 \\ -1.44 \\ 1.13 \\ 0.50 \\ 0.38 \\ -0.36 \\ -0.57 \\ 1.20 \\ 0.67 \\ 0.78 \\ 1.10 \\ -1.80 \end{pmatrix}$$

Matriz 2: Pesos y ganancias correspondiente a la segunda capa.

$$\mathbf{W}^{32} = \begin{pmatrix} -0.43 & -0.52 & -0.90 & 0.32 & 0.05 & -0.79 & 0.65 & 0.80 & 0.51 & 0.77 & -0.44 & 0.43 \end{pmatrix}$$

$$\mathbf{b}_3 = \begin{pmatrix} 0.45 \end{pmatrix}$$

Matriz 3: Pesos y ganancias correspondiente a la tercera capa.

$$\mathbf{W}^{11} = \begin{pmatrix} 0.84 & 0.72 \\ -0.65 & 1.38 \\ -0.22 & 0.49 \\ -1.01 & -0.04 \\ -0.85 & 1.53 \\ 0.71 & 0.60 \\ -0.91 & 0.31 \\ -0.93 & 0.53 \\ 0.93 & 0.40 \\ 0.56 & -1.40 \\ -1.01 & -0.76 \\ -1.02 & 1.23 \\ 0.74 & -0.50 \\ -1.00 & -0.34 \\ -0.78 & -0.13 \\ -0.46 & -0.31 \end{pmatrix}; \qquad \mathbf{b}_1 = \begin{pmatrix} -1679.8 \\ 1293.2 \\ 469.7 \\ 2020.8 \\ 1699.9 \\ -1416.2 \\ 1827.0 \\ 1861.6 \\ -1864.5 \\ -1109.5 \\ 2032.6 \\ 2034.1 \\ -1483.9 \\ 2005.8 \\ 1560.1 \\ 919.7 \end{pmatrix}$$

Matriz 4: Pesos y ganancias correspondientes a la primera capa de la salida óptima.

$$\mathbf{W}^{21} = \begin{pmatrix} -0.52 & -0.06 & -0.45 & 0.19 & -0.84 & -0.03 & 0.11 & -0.62 & -0.48 & 0.25 & -0.23 & 0.41 & 0.66 & -0.91 & -0.93 & -0.48 \\ -0.19 & 0.07 & -0.53 & 0.35 & -0.91 & 1.24 & 0.36 & 0.52 & 0.14 & -0.86 & -0.17 & -0.15 & 0.17 & 0.14 & -0.23 & 0.23 \\ -0.14 & 0.5 & -0.15 & 0.62 & -0.45 & -0.40 & 0.65 & 0.31 & 0.51 & 0.21 & -0.38 & 0.07 & -0.62 & -1.11 & 1.34 & 1.01 \\ -0.86 & -0.03 & 0 & 1.42 & 0.78 & 0.60 & 0.90 & 0.8 & 0.22 & 0.37 & -0.15 & -0.2 & -0.22 & 1.02 & 0.07 & 0.84 \\ -0.12 & -0.21 & -0.23 & -0.12 & 0.86 & 0.01 & 0.98 & 0.51 & 0.81 & 0.10 & 0.16 & -0.70 & -0.28 & -0.67 & 0.37 & -0.87 \\ 0.05 & -0.76 & 0.53 & 0.30 & 0.28 & 1.22 & -0.21 & 0.41 & 0.09 & -0.60 & 0.52 & -0.55 & 0.80 & -0.35 & 1.13 & 0.56 \\ -0.28 & -0.31 & -0.31 & 0.00 & 0.14 & -0.85 & 0.54 & 0.20 & -0.25 & -0.52 & -0.36 & 0.58 & -0.27 & -0.06 & 0.79 & 0.88 \\ 0.92 & 0.01 & 0.81 & 0.13 & -0.48 & 0.17 & -0.15 & 0.72 & -0.39 & 0.11 & 0.46 & -0.89 & 0.95 & -1.16 & 0.24 & -0.55 \\ 0.30 & 0.09 & -0.24 & 0.34 & -0.19 & -0.34 & 0.46 & -0.68 & 0.02 & -0.95 & 1.23 & 0.39 & 0.44 & 0.05 & -0.92 & -0.91 \\ 0.97 & -0.37 & -0.49 & 0.49 & -0.04 & 1.39 & 0.23 & 0.76 & 0.12 & -1.03 & 0.28 & -0.41 & 0.89 & 0.32 & 0.69 & 0.81 \\ 0.47 & 0.57 & 0.25 & -1.45 & 0.88 & -0.65 & -0.47 & 0.20 & 0.06 & -0.36 & -0.14 & 0.86 & 0.63 & 1.47 & -0.07 & -0.49 \\ -0.03 & 0.31 & -1.02 & -0.12 & 0.80 & 1.75 & -0.24 & -0.19 & 0.02 & 0.14 & 0.09 & -0.02 & -0.14 & -0.03 & -0.77 & 0.24 \end{pmatrix}$$

$$\mathbf{b}_2 = \begin{pmatrix} 1.63 \\ -1.42 \\ 1.12 \\ 0.51 \\ 0.38 \\ -0.36 \\ -0.57 \\ 1.20 \\ 0.67 \\ 0.78 \\ 1.10 \\ -1.80 \end{pmatrix}$$

Matriz 5: Pesos y ganancias correspondiente a la segunda capa.

$$\mathbf{W}^{32} = \begin{pmatrix} -0.44 & -0.52 & -0.89 & 0.31 & 0.06 & -0.80 & 0.67 & 0.83 & 0.53 & 0.77 & -0.44 & 0.45 \end{pmatrix}$$

$$\mathbf{b}_3 = \begin{pmatrix} 0.43 \end{pmatrix}$$

Matriz 6: Pesos y ganancias correspondiente a la tercera capa.

Bibliografía

[1] Mutasem Khalil Sari Alsmadi, Khairuddin Bin Omar, and Shahrul Azman Noah. Back propagation algorithm: The best algorithm among the multi-layer perceptron algorithm. *International Journal of Computer Science and Network Security*, 9(4):378–383, Abril 2009.

[2] Bruce L. Bowerman, Richard T. O'Connell, and Anne B. Koehler. *Pronósticos, Series de Tiempo y Regresión. Un Enfoque Aplicado.* Cengage Learning, 2009.

[3] Peter J. Brockwell and Richard A. Davis. *Introduction to Series and Forecasting.* Springer, 2002.

[4] Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods.* Statistics. Springer, second edition, 2006.

[5] Maria Isabel Acosta Buitrago and Camilo Alfonso Zuluaga Muñoz. *Tutorial Sobre Redes Neuronales Aplicada en Ingeniería Eléctrica y su implementación en un sitio Web.* Universidad Tecnológica de Pereira. Facultad de Ingenieria Electrica, 2000.

[6] Howard Demuth, Mark Beale, and Martin Hagan. *Neuronal Network Toolbox 5. User's Guide.* The Math Works, 2007.

[7] Víctor M. Guerrero G. *Modelos Estadísticos para Series de Tiempo Univariadas.* Departamento de Matemáticas, Centro de Investigación y de Estudios Avanzados del IPN, 1987.

[8] Martin T. Hagan, Howard B. Demuth, and Mark Beale. *Neural Network Design.* PWS Publishing Company, 1996.

[9] Ali Haydar, Zafer Agdelen, and Pinar Özbeseker. The use of backpropagation algorithm in the estimation of the firm performance. *Ístanbul Ticaret Üniversititesi Fen Bilimleri Dergisi Yil*, pages 51–64, 2006.

[10] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Pearson Prentice Hall, segunda edition, 1999.

[11] José R. Hilera and Victor J. Martinez. *Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones*. Alfaomega, 2000.

[12] T. Jayalakshmi and Dr. A. Santhakumaran. Improved gradient descent backpropagation neural networks for diagnoses of type ii diabetes mellitus. *Global Journal of Computer Science and Technology*, 9(5):94–97, Enero 2010.

[13] M. Asunción Gonzalez S. Williams Gómez López José del C. Jiménez H. Demanda de energía eléctrica: Un análisis usando series de tiempo. In *Memorias de la Tercera Semana Internacional de Estadística y Probabilidad*, 2010.

[14] Fidel Ernesto Hernández Montero and Wilfredo Falcón Urquiaga. Cancelación de ruido a través de técnicas neuronales. In *IV Congresso Brasileiro de Redes Neurais*, pages 1–6, Sao José dos Campos, Brazil, Julio 1999. IV Brazilian Conference on neural networks.

[15] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications Whit R Examples*. Springer, 2006.

[16] R Development Core Team. R: A language and environment for statistical computing, 2010.

[17] Iván Cruz Torres. *Pronóstico en el Mercado de Derivados utilizando Redes Neuronales y Modelos ARIMA: una aplicación al Cete de 91 días en el MexDer*. PhD thesis, Facultad de Contaduría y Administración. Universidad Nacional Autónoma de México, 2007.