CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO
DEPARTAMENTO DE CONTROL AUTOMÁTICO

Identificación de series de tiempo utilizando el método Bayesiano.

TESIS

Que presenta

Jorge de Jesus Morales Mercado

para obtener el Grado de

DOCTOR EN CIENCIAS

EN LA ESPECIALIDAD DE

CONTROL AUTOMÁTICO

Directores de la Tesis:

Dr. Wen Yu Liu
Dr. Floriberto Ortiz Rodríguez

Ciudad de México Marzo 2022

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) y al Centro de Investigación y Estudios Avanzados (CINVESTAV), por el apoyo económico durante de mis estudios de posgrado con el cual se pudo lograr el desarrollo y objetivos de este trabajo.

A mis padres Maria Elena Mercado Murillo y Vicente Morales Cardoso, por su apoyo incondicional que siempre me han brindado, dedico este trabajo con todo mi corazón.

A mis amigos por su apoyo y amistad en todos los momentos.

A mis directores de tesis el Dr. Wen Yu Liu y el Dr. Floriberto Ortiz Rodríguez, por su apoyo y guía para el desarrollo de este trabajo.

Sin ustedes este trabajo no sería posible.

A todos, Muchas Gracias.

Resumen

Los avances del aprendizaje automático permiten desarrollar estructuras y combinaciones de redes neuronales artificiales que facilitan y mejoran el rendimiento en las tareas a las que se aplican. En este trabajo se genera una estructura que combina la información disponible con la información estadística de los datos de entrada-salida en el sistema dinámico, a partir de la cual se puede generar una estructura de ajuste fino de pesos con dos lotes de entrenamiento, el primer lote entrena la red neuronal con la información de entrada y salida disponible mientras que el segundo lote utiliza la información estadística para actualizar de nuevo los pesos en la red neuronal. Por otro lado, una red neuronal se aplica para modelar distribuciones de probabilidad combinadas con la inferencia bayesiana para obtener distribuciones de probabilidad a posteriori que pueden aplicarse a información importante como pueden ser los parámetros sísmicos.

Abstract

Machine learning advances allow to development structures and combinations of artificial neural networks that facilitate and improve the performance on tasks to which they are applied. In this work we generate a structure that combines available information with the statistical information of input-output data in the dynamic system, from which a fine tuning structure of weights can be generated with two training batches, the first batch trains the neural network with the available input and output information while the second batch uses the statistical information to update again the weights in the neural network. On the other hand, a neural network is applied to model probability distributions combined with Bayesian inference to obtain posterior probability distributions that can be applied to important information such as seismic parameters.

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

Las metodologías de inferencia Bayesiana surgen como herramienta de aprendizaje automático por sus facilidades de crear conocimientos y actualizaciones en el enfoque probabilístico de los datos, siendo una herramienta similar a las redes neuronales artificiales. Incluso estas herramientas se han implementado y asimilado dentro de los procesos de aprendizaje automático, donde la diferencia principal con las redes neuronales artificiales, que minimizan el error de modelado en la salida de la red con los datos reales analizados para actualizar los pesos en una red, en la metodología Bayesiana en redes neuronales busca una distribución de probabilidad en los pesos de la red aplicando la teoría Bayesiana para identificar una distribución de probabilidad que se tiene en los datos y que es llamada como predicción de la distribución.

Los aprendizajes de redes neuronales con enfoque Bayesiano se han presentado en los años ochenta sumando esta alternativa a la identificación de sistemas dinámicos obteniendo resultados para los datos representados mediante información de probabilidad como lo es una distribución. Por lo que estas redes neuronales que podemos llamar de tipo Bayesiano se utilizan principalmente para obtener clasificación, identificación y predicción de distribuciones de probabilidad.

Con los resultados que arrojan los enfoques de aprendizaje en las redes neuronales y los obtenidos con las redes neuronales Bayesianas, se busca realizar redes neuronales con

estructuras profundas que combinen estos resultados y mejoren el rendimiento al momento de aplicar la red neuronal a datos reales.

1.1. Motivación

En este trabajo se busca realizar la combinación de las metodologías Bayesianas y las redes neuronales para generar una estructura profunda que nos permita realizar los procedimientos de inferencia Bayesiana para distribuciones posteriores, así como una estructura que combinen estos enfoques para la identificación de sistemas dinámicos.

1.2. Objetivos

Objetivo General:
Generar una estructura con redes neuronales para la identificación de sistemas dinámicos mediante la combinación de información estadística disponible en el sistema analizado.

Objetivos particulares:

- Utilizar la inferencia Bayesiana para la obtención de distribuciones a posteriori por medio de información reciente.
- Aplicar la inferencia Bayesiana para el modelado de la distribución de probabilidad de parámetros sísmicos.
- Combinar una red neuronal y la metodología Bayesiana para la obtención distribuciones de probabilidad a posteriori.
- Combinar redes neuronales con información estadística para la identificación de sistemas dinámicos.

1.3. Estructura

En este trabajo se analiza la teoría Bayesiana así como la aplicación de esta en redes neuronales, desarrollando en el capítulo 2 el teorema de Bayes para obtener distribuciones de probabilidad además de las estructuras principales de las redes neuronales hasta definir una estructura profunda y aprendizaje profundo para una red neuronal, el capítulo 3 describe el análisis sobre el cálculo de distribuciones de probabilidad posteriores y una mejora en el proceso de actualización de distribuciones por medio de datos recientes realizando una implementación sobre distribuciones de probabilidad para parámetros sísmicos, el capítulo 4 realiza la combinación de las redes neuronales y el método Bayesiano para la obtención de distribuciones posteriores, así como, una estructura para obtener identificar sistema dinámicos mediante la información estadística disponible, en el capítulo 5 se tienen la aplicación del cálculo de distribuciones a posteriori para parámetros sísmicos así como la identificación de dos sistemas dinámicos, mientras que en los capítulo 6 y 7 se presentan las conclusiones y trabajos a futuro respectivamente.

1.4. Contribuciones

La primera parte de este trabajo se enfoca en el análisis y desarrollo de técnicas basadas en distribuciones de probabilidad para la búsqueda de distribuciones que reflejen información de eventos que ocurren en el futuro. Debido a las facilidades que presentan las teorías Bayesianas para el cálculo de distribuciones posteriores y su analogía con resultados predictivos que realiza una red neuronal, se utilizan estas herramientas en este trabajo para obtener un análisis y resultados sobre las distribuciones a posteriori por medio de información reciente en el cálculo de la inferencia Bayesiana. Se estudió e implementó el método de Monte Carlo para obtener las distribuciones a posteriori de los parámetros de magnitud, distancia y tiempo en los sismos en una región de Italia y México, además implementar información reciente para el cálculo de las distribuciones obteniendo mejoras en la predicción de estas.

Con esto se presentaron dos trabajos en conferencias internacionales:

- Morales, J., Yu, W., & Telesca, L. (2019). Bayesian Analysis of the Magnitude of Earthquakes Located in a Seismic Region of Italy. MDPI Proceeding, Vol.24(1), DOI:10.3390/IECG2019-06214.
- Morales, J., & Yu, W. (2021, October). A novel Bayesian inference based training method for time series forecasting. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 909-913). IEEE. DOI: 10.1109/SMC52423.2021.9659009.

Una vez que se plantearon las bases de los enfoques de probabilidad para la predicción de las distribuciones, se realizaron combinaciones de este enfoque con las redes neuronales artificiales con el objetivo de obtener distribuciones a posteriori. También se obtuvo una estructura basada en redes neuronales que permite identificar sistemas dinámicos mediante la información estadística disponible en el sistema dinámico analizado.

Con estos enfoques se publicaron los siguientes trabajos en revistas especializadas:

- Morales, J., Yu, W., & Telesca, L. (2020). Bayesian Approach for Estimating the Distribution of Magnitudes, Interevent Times and Distances of Earthquake Sequences. Cybernetics and Systems, 51(8), 733-745. DOI:10.1080/01969722.2020.1814582.
- Morales, J., Yu, W., & Telesca, L. (2022). Bayesian inference for data-driven training with application to seismic parameter prediction. Soft Computing, 26(2), 867-876. DOI:10.1007/s00500-021-06232-z.
- Morales, J., & Yu, W. (2021). Improving neural network's performance using Bayesian inference. Neurocomputing, 461, 319-326. DOI:10.1016/j.neucom.2021.07.054.

Capítulo 2

Redes neuronales y enfoque Bayesiano para modelado de datos.

En este trabajo se desarrolla la teoría y aplicación de redes neuronales considerando el empleo de información estadística en el modelado de sistemas dinámicos e implementación de redes neuronales Bayesianas para el modelado de distribuciones de probabilidad, con esto se busca modelar sistemas dinámicos además de obtener distribuciones de probabilidad sobre parámetros del mismo sistema. Se presentan estructuras que complementan a las redes neuronales simples con las cuales se busca la mejora en el rendimiento en la capacidad de identificación de sistemas dinámicos por medio de la red neuronal.

En los años cincuenta, Frannk Roseblant implementó ideas presentadas por McCulloch y Pitts (1943) para implementar ordenadores que fungieran como redes neuronales artificiales, donde apareció la estructura de una red neuronal artificial. En este trabajo se aplica una estructura simple que simula la neurona artificial cerebral llamada "perceptrón", con la cual se pueden realizar tareas de clasificación sencilla como lo es una tarea de tipo seleccionador "OR" donde se elige una sección u otra, así como, una tarea de seleccionador "AND" donde se determinan conjuntos de una sección. Pero al presentarse tareas como un seleccionador de tipo "XOR" este tipo de red neuronal tiene complicaciones [25]. En la siguiente figura 2.1

se observa la estructura de una red neuronal artificial simple con sólo una capa de pesos con una función de activación, además de que el número que se indica en las redes neuronales artificiales para sus capas esta dado por el número de capas de pesos que esta contiene.

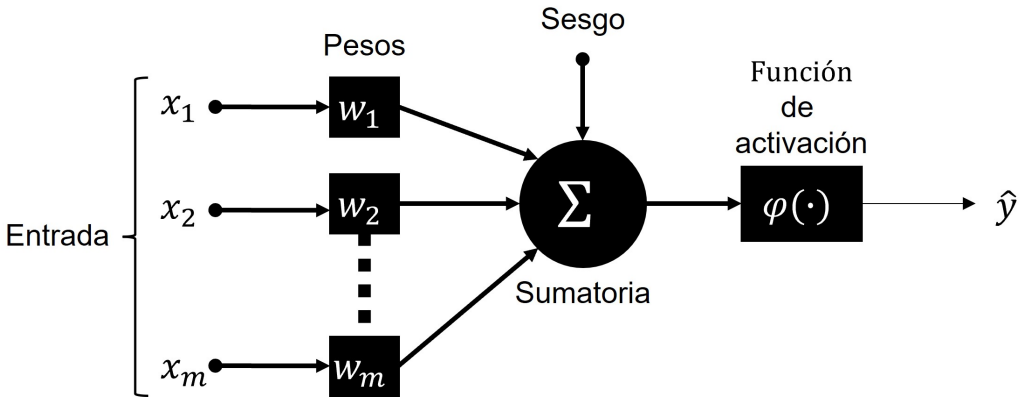


Figura 2.1: Modelo de una red neuronal con una Neurona (Perceptrón).

La red neuronal artificial tiene como salida y que es la sumatoria de las entradas x al cual se le otorga un índice llamado peso ω y que pasa por medio de una función de activación φ:

$$\hat{y} = \varphi\left(\sum_{i=1}^{i=m} W_i x_i\right) \tag{2.1}$$

En la estructura de una red neuronal se modifica el número de sus capas para cumplir con la tarea asignada, obteniendo una profundidad asignando un numero de capas en la red neuronal de forma horizontal y el aumento de neuronas por capa de forma vertical, se considera el número de capas en la red como el número de pesos en la red y no el número de capas de neuronas:

$$\hat{y} = \varphi\left(\sum_{i,j=1}^{i,j=m} W_{ji} x_i\right) \tag{2.2}$$

La estructura que contiene más de una capa es conocida como perceptrón multicapa y a estas modificaciones en la estructura para hacerla más grande se también lleva el nombre de

estructura profunda para la red neuronal. Los principales cambios en la estructura profunda de una red se llevan a cabo por medio de más capas ocultas en la estructura u otra red neuronal conectada en serie para la modificación en horizontal y la profundidad en la estructura vertical puede llevarse a cabo mediante el aumento de nodos en las capas ocultas de la red u otra red neuronal conectada en paralelo.

El entrenamiento de la red se basa en el error modelado el cual se define como la diferencia entre la salida del sistema dinámico y la salida de la red que lo modela. Este error se emplea en una función de costo J como se observa en la ecuación (2.3), la cual se debe minimizar mediante la modificación de los pesos de la red neuronal. Para este caso, como se identifica un sistema dinámico con las variables x como la entrada y y como la salida se tomará una nueva nomenclatura para la red como u para la entrada a la red y $\hat{y}$ será la salida de la red:

$$J(k) = \frac{1}{2}e^2 = \frac{1}{2}(y(k) - \hat{y}(k))^2 \tag{2.3}$$

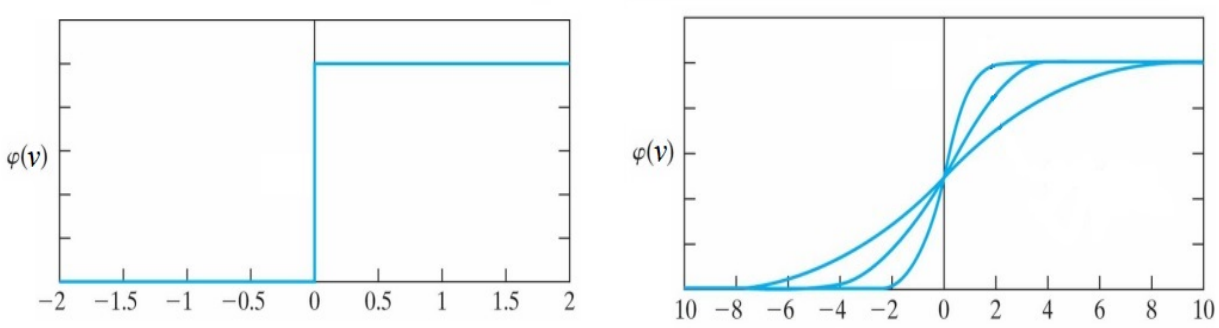
Para optimizar los pesos se utiliza la metodología de entrenamiento conocida como propagación hacia atrás del error (Back-propagation), en la cual el error cuando se propaga a la capa oculta en la red representa la derivada del error analizada en la salida con relación a cada uno de los pesos W_i:

$$\frac{\partial J}{\partial W_i} = \frac{\partial \hat{y}}{\partial W_i}\frac{\partial J}{\partial \hat{y}} = \frac{\partial \hat{y}_i}{\partial \varphi}\frac{\partial \varphi}{\partial W_i}\frac{\partial J}{\partial \hat{y}} \tag{2.4}$$

Por lo que la actualización de los pesos queda representada en función del error de modelado que aparece en la capa de pesos de la red neuronal analizado mediante una taza de aprendizaje (η):

$$\Delta\omega_i = -\eta\frac{\partial J}{\partial W_i} \tag{2.5}$$

La última parte de la estructura de la red neuronal es procesada por una función de lógica aplicada a la salida de la red neuronal, quedando clasificados los valores que salen de la red neuronal, y que se conoce como función de activación que puede tener valores binarios o en la mayoría de los casos se encuentra con una forma sigmoide.



(a) Función Binaria (b) Función Sigmoide

Figura 2.2: Funciones de activación.

En el caso de la salida de la red neuronal se tiene una función que transmite la información de las partes internas de la red neuronal hacia la información de salida de la red, estas funciones pueden corresponder a un filtro, un límite o un umbral para los datos, con los cuales se transmite la información de la red que proviene de las sumatoria de los pesos y entradas (ωx) hacia la salida de esta. Las funciones de activación de tipo sigmoidal más comunes:

- Función logística o escalón entre cero y uno [0,1]:

$$\varphi(v) = \frac{1}{1 + e^{-av}}$$

- Función tangente hiperbólica entre menos uno y uno [-1,1]:

$$\varphi(v) = \tanh(v) = \frac{e^{v} - e^{-v}}{e^{v} + e^{-v}}$$

Donde v es la entrada a la función de activación como $v = \sum \omega_m x_m$.

Aunque también se pueden encontrar otras funciones de activación en las redes neuronales que también son utilizadas aunque se emplean con menor frecuencia.

- función identidad con rango [-∞,∞]:

$$\varphi(v) = (v)$$

- función Gaussiana con rango [0,1]:

$$\varphi(v) = Ae^{Bv}$$

- función sinusoidal con rango [-1,1]:

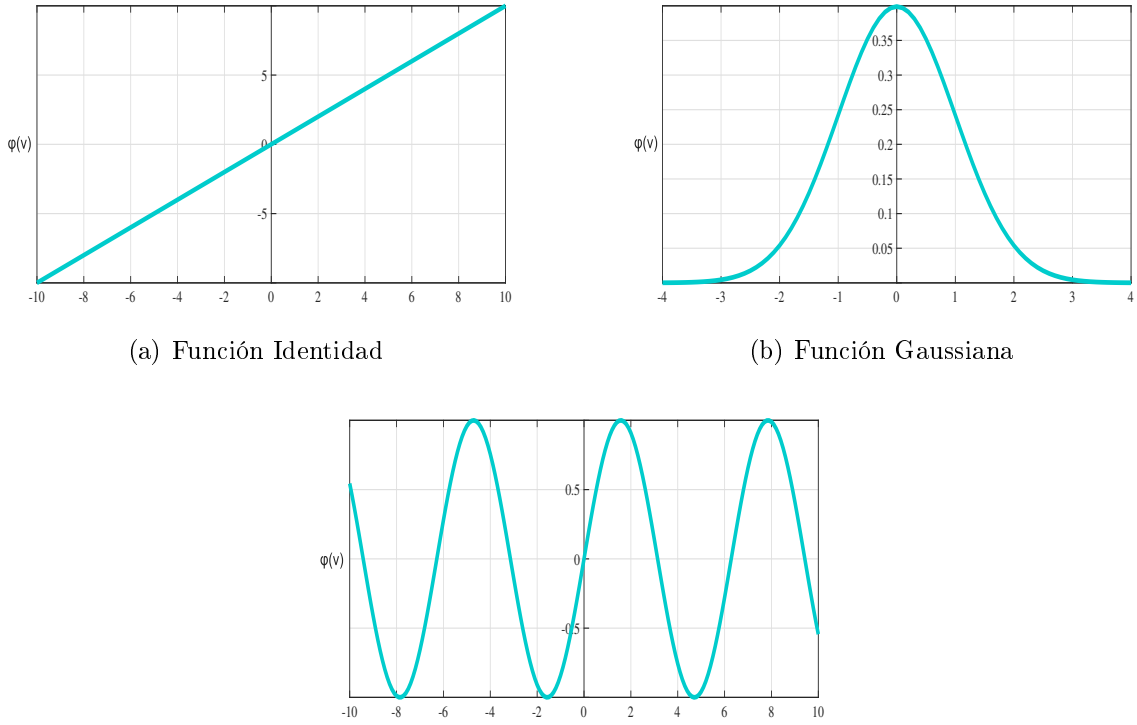
$$\varphi(v) = Asen(\varpi v + \varrho)$$

donde A, ϖ y ϱ representan la amplitud, la frecuencia angular y la fase respectivamente.

En la figura 2.3 se muestran las funciones de activación para la red neuronal que se usan con menos frecuencia.

2.1. Redes neuronales con aprendizaje profundo.

En esta sección se desarrollan los principales puntos teóricos y prácticos cuando una red neuronal se complementa por medio de una estructura de tipo profunda, enfocando en el tipo de análisis que se ocupa para el aprendizaje automático *(Machine Learning)*. Se encuentran dos principales características para generar una estructura profunda, generando una estructura horizontal aumentando el número de capas en la red neuronal, o una estructura vertical donde se aumenta el número de nodos en la red neuronal. Estas estructuras aparecen debido a que se desea mejorar el rendimiento de las redes neuronales ante las tareas a realizar evitando que las redes neuronales crezcan demasiado en la cantidad de neuronas en la capa oculta aunque el teorema de aproximación universal nos lo permita [31] pero recordando el problema de sobre ajuste en la cual la etapa de entrenamiento se ajusta



(a) Función Identidad

(b) Función Gaussiana

(c) Función Sinusoidal

Figura 2.3: Funciones de activación.

muy bien mientras que la prueba de la red con el conjunto de datos de prueba tiene malos resultados [70].

2.1.1. Aprendizaje en redes neuronales con múltiples capas (*Deep Learning*).

El aprendizaje profundo permite realizar modelos de redes neuronales que están compuestos de múltiples capas de procesamiento. Estos métodos han obtenido buenos resultados en reconocimiento de voz, reconocimiento de objetos visuales, detección de objetos entre otras tareas de la ingeniería [40]. El aprendizaje profundo desarrolla estructura compleja

en grandes conjuntos de datos mediante el uso del algoritmo de propagación hacia atrás del error (*Back-Propagation*) para indicar cómo una máquina debe cambiar sus parámetros internos que se tienen en la estructura de la red. Un gran ejemplo de estas estructuras como las redes neuronales convoluciones (*CNN*) profundas han mejorado el rendimiento en los campos de procesamiento de imágenes, video, voz y audio, así como, estructuras de redes neuronales profundas de tipo recurrente se han aplicado en análisis de texto y el habla obteniendo buenos resultados [40].

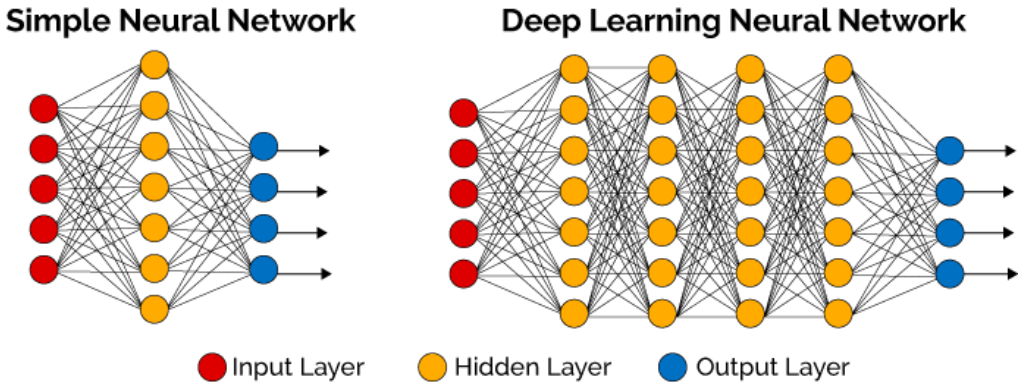


Figura 2.4: Red Neuronal vs Red neuronal de múltiples capas [40].

Aprendizaje Supervisado con múltiples capas.

El aprendizaje supervisado es la herramienta más común para la implementación de las estructuras profundas en una red neuronal [40]. Para la ejecución de este tipo de aprendizaje se debe contar con un grupo de datos disponibles catalogados en cada una de sus especificaciones, como la entrada y salida de un sistema.

El entrenamiento en la red ajusta sus parámetros, como lo son los pesos ω, para otorgar los mejores resultados de la tarea a realizar, como lo puede ser en la identificación de un sistema dinámico el comportamiento en la salida de este sistema. Recordando que el ajuste de los parámetros por medio del entrenamiento de la red se realiza mediante el error de

modelado $e = y - \hat{y}$ y minimizando una función de costo basado en este mismo error. Debido a que nos encontramos con estructuras que aumentan la cantidad de capas ocultas en estas se ajustan un mayor número de parámetros debido a que cada una de las capas que se aumenta tienen sus propios pesos.

El objetivo principal en la función de costo es reducir el error de modelado en la red neuronal mediante el cambio de los pesos en cada una de las capas de la estructura en la red, a este proceso le conoce como gradiente descendente. Por lo general este proceso se aplica en pequeños pasos hasta que el promedio de la función objetivo deja de disminuir. Una vez que se obtiene la estructura entrenada se pone a prueba con un conjunto de datos que se desconocen para evaluar su rendimiento.

Una de las principales razones y ventajas por las cuales se implementan estas estructuras es que son insensibles a grandes variaciones en el conjunto de datos como el fondo, la pose, iluminación y objetos circundantes en la identificación de una imagen [40], o como la perdida de datos y hasta el ruido en un sistema dinámico.

2.1.2. Entrenamiento de Redes con múltiples capas.

Las estructuras con múltiples capas en las redes neuronales generalmente se entrenan por medio del método de gradiente descendente aplicando el procedimiento propagación hacia atrás del error de modelado.

La aplicación del método de gradiente descendente para una red neuronal con dos o más capas de neuronas ocultas se realiza mediante la regla de la cadena para derivadas que contienen el error de modelado que se distribuye a cada una de las capas en la estructura de la red neuronal. Con lo que la aplicación de la propagación del error hacia atrás puede ser aplicada a estructuras con muchas capas de neuronas ocultas con el objetivo de llevar el análisis del error “propagar”, desde la salida en el sistema dinámico analizado hasta llegar a

la capa donde se ingresan los datos a la red.

En la figura 2.5 se puede observar una red neuronal con múltiples capas en su estructura con la aplicación del entrenamiento de retro propagación. Donde se muestra a) Separación de las regiones de entrada. b) Regla de la cadena para derivadas. c) Ecuaciones para realizar el paso hacia adelante en una red. d) Ecuaciones para realizar un paso hacia atrás en una red.

Debido a la aplicación de estas metodologías en el aprendizaje de redes con estructura profunda se sigue teniendo el inconveniente de encontrarse con un mínimo local en el error o un punto silla, que no se consideran grandes problemas y que no son el objetivo de análisis de este trabajo.

Para la profundidad de las capas se adapta el proceso de entrenamiento de propagación hacia atrás por medio las ecuaciones del error cuadrático medio $J = (y - \hat{y})^2$ sus derivadas se obtienen con la aplicación de regla de la cadena.

La salida de la red puede representarse como:

$$\hat{y} = \varphi_1 \left(\sum_{i=1}^{m} W_i a_i + b_1 \right) = \varphi_1(z) \tag{2.6}$$

donde $a_i = \varphi_2(\sum_{j=1}^{m} V_{ij} x_j + b_2)$ representa a una capa oculta, b_1 es el sesgo que se tiene en la salida red neuronal, φ_1 y φ_2 son las funciones de activación en la red, con $i = (1, .., m)$ debido a que m es el total de neuronas. Con lo que la función de costo a minimizar puede representare en función de los parámetros de la red neuronal:

$$J = \left(y - \varphi_1 \left(\sum_{i=1}^{m} W_i a_i + b_1 \right) \right)^2 \tag{2.7}$$

Para simplificar vamos a sustituir los parámetros en la red como $z = \sum_{i=1}^{m} W_i a_i + b_1$ obteniendo a la red neuronal $\hat{y} = \varphi_1(z)$, con lo que se obtiene la salida de la red dependiendo de cada uno de los pesos W_i:

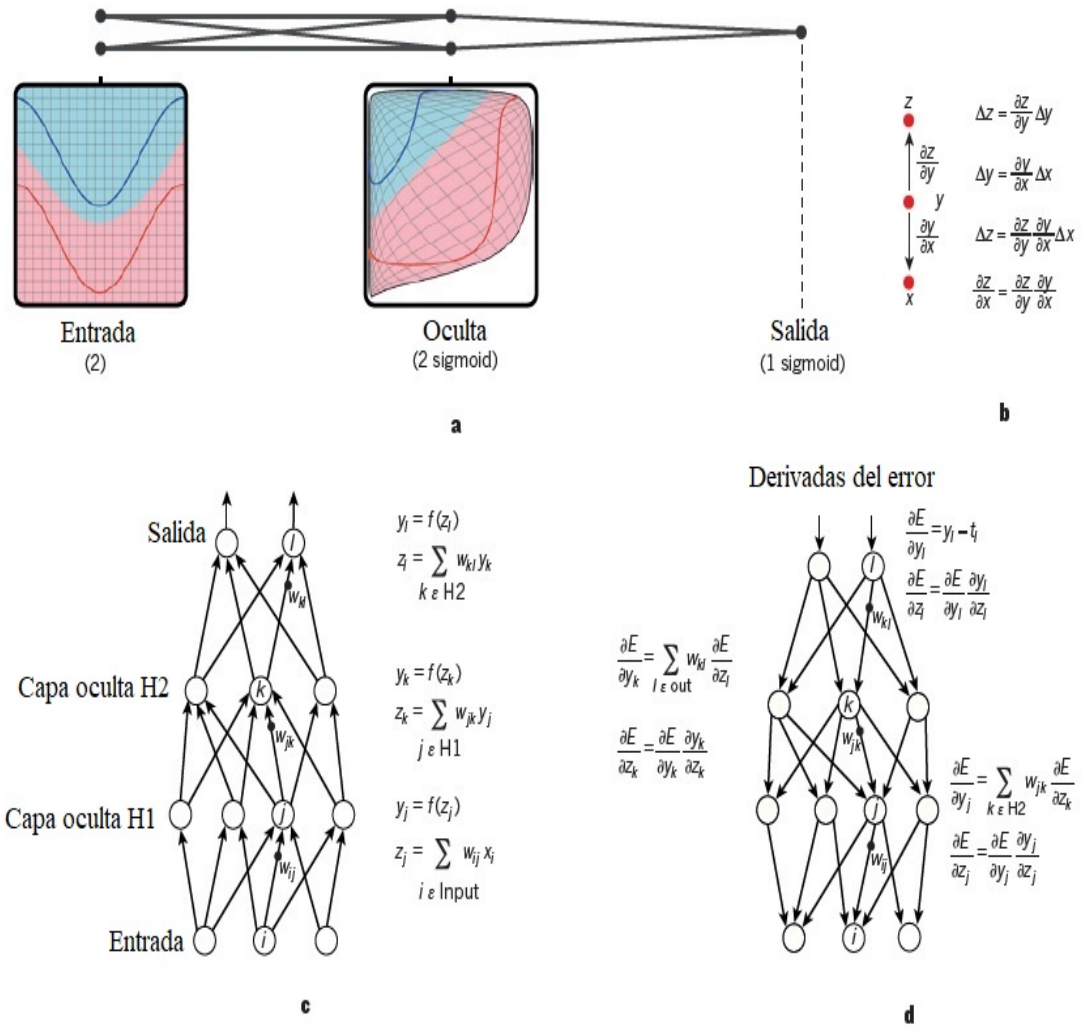


Figura 2.5: Aprendizaje de una red con múltiples capas [40].

$$\frac{\partial \hat{y}}{\partial W_i} = \frac{\partial \hat{y}_i}{\partial \varphi_1} \frac{\partial \varphi_1}{\partial z} \frac{\partial z}{\partial W_i} \tag{2.8}$$

Además de la relación que se tiene en la salida con respecto de los sesgo en la red neuronal:

$$\frac{\partial \hat{y}}{\partial b_1} = \frac{\partial \hat{y}_i}{\partial \varphi_1} \frac{\partial \varphi_1}{\partial z} \frac{\partial z}{\partial b_1} \tag{2.9}$$

Para propagar el error en las capas ocultas de la red se utilizan las siguientes expresiones:

$$\begin{aligned} \frac{\partial J}{\partial a_i} &= W_i \frac{\partial J}{\partial z} \\ \frac{\partial J}{\partial r_i} &= \frac{\partial J}{\partial a_i} \frac{\partial a_i}{\partial r_i} \end{aligned} \tag{2.10}$$

donde r_i representa a la combinación de las entradas y pesos de la capa oculta mas el sesgo de esta capa $r_i = V_{ij} x_j + b_2$:

$$\begin{aligned} \frac{\partial J}{\partial x_j} &= \sum_{i=1}^{m} V_{ij} \frac{\partial J}{\partial r_i} \\ \frac{\partial J}{\partial V_{ij}} &= \sum_{i=1}^{m} \frac{\partial J}{\partial r_i} \frac{\partial r_i}{\partial V_{ji}} \end{aligned} \tag{2.11}$$

La actualización de los pesos de la capa de salida y de la capa oculta contigua se representa de la siguiente forma:

$$\begin{aligned} \Delta W_i &= -\eta \frac{\partial J}{\partial W_i} \\ \Delta V_{ij} &= V_{ij} - \eta \frac{\partial J}{\partial V_{ij}} \end{aligned} \tag{2.12}$$

Si la estructura tiene más capas se sigue el mismo procedimiento considerando los pesos y las variables correspondientes en cada capa.

2.1.3. Tipos de aprendizaje en la redes neuronales

Las redes neuronales contienen parámetros que se ajustan para realizar una actividad como lo puede ser la clasificación o la identificación. A los algoritmos que realizan estas tareas se les conoce como métodos de aprendizaje y se clasifican en tres grupos:

- Aprendizaje Supervisados.
- Aprendizaje No supervisados.
- Aprendizaje por refuerzo.

Aprendizaje supervisado.

En los métodos de entrenamiento supervisado se cuenta con un instructor que muestra los valores de los parámetros y además indica las acciones correspondientes para ajustar los parámetros de la red neuronal. Estos algoritmos pueden involucrar dos partes en su funcionamiento: El entrenamiento que corresponde al ajuste de los parámetros de la red para generar un modelo adecuado de esta, y la parte de prueba que es la ejecución de la red sobre los datos a identificar o clasificar. En este tipo de aprendizaje se encuentran algunos algoritmos habituales para actualizar los pesos en las redes neuronales como los son:

- Algoritmo de regresión (Back propagation).
- Regresión por Mínimos Cuadrados.
- Arboles de decisión.

Además de que existen estructuras con algoritmos de aprendizaje supervisado como lo son:

- Support Vector Machine (SVM).
- Máquinas Restringidas de Boltzmann (RBM).

En la figura 2.6 se puede observar la estructura que se sigue para generar una red neuronal artificial con un algoritmo de aprendizaje supervisado, en la imagen se presenta una estructura que tiene como objetivo identificar un grupo de imágenes las frutas.

Los algoritmos de aprendizaje para las redes neuronales más comunes son:

Red de memoria a largo plazo: Estas redes están compuestas por capas de memoria [28]. Son llamadas redes de memoria a largo plazo o *Long short-term memory (LSTM)* por sus siglas en inglés, y cada una de las células de la memoria está basada en compuertas con diferentes funciones, como lo es la compuerta de entrada que tiene como función la escritura

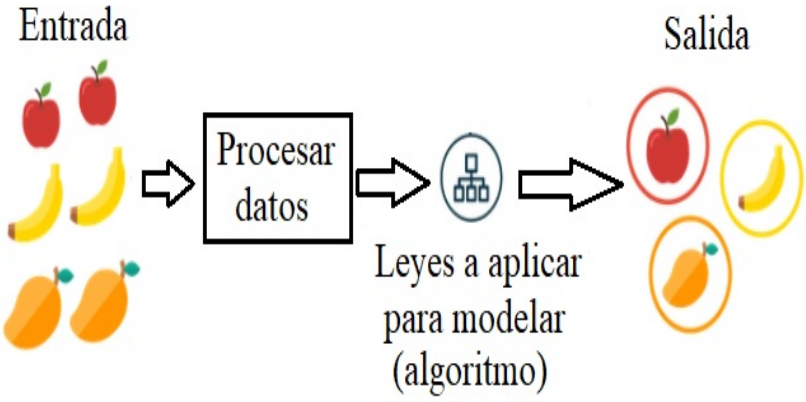


Figura 2.6: aprendizaje supervisado.

en la celda, la compuerta de salida que funciona como lectura, la compuerta de olvido que descarta información no necesaria para la celda, estas partes de la red de largo plazo y su estructura se puede observar en la figura 2.7 con h_t unidades de celda para cada unidad de memoria en la red.

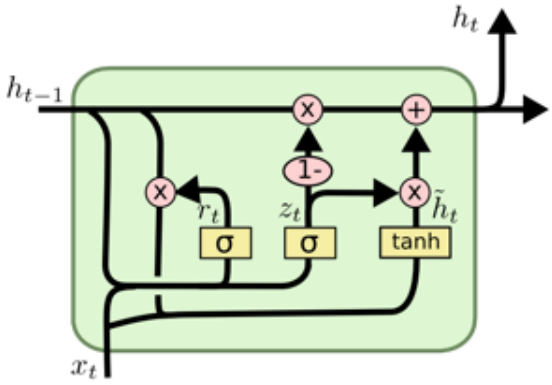


Figura 2.7: Arquitectura de la celda de memoria para una red *LSTM* [69].

En el caso de presentar una conexión entre celdas la red de memoria a largo plazo se conecta en serie. Se puede observar la conexión de 3 células de memoria en la red de memoria a largo plazo en la figura 2.8.

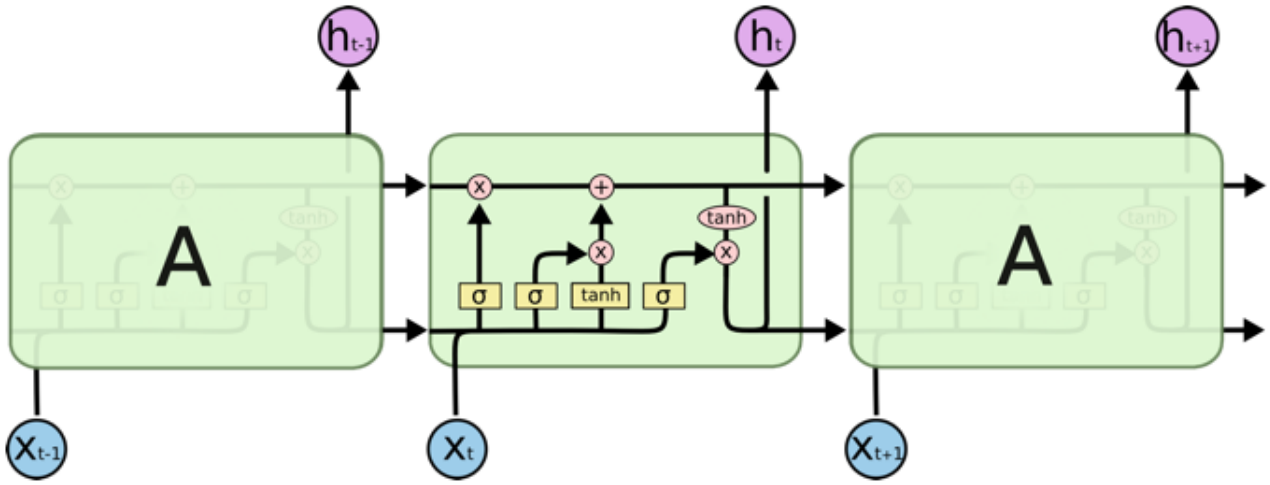


Figura 2.8: Conexión de 3 células de memoria de una *LSTM* [69].

Para esta estructura se puede generar un algoritmo con el cual se entrenan las células de memoria de las redes de memoria a largo plazo, el cual corresponde a las siguientes ecuaciones:

$$i_t = \sigma_i(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2.13}$$

$$f_t = \sigma_f(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2.14}$$

$$c_t = f_t c_{t-1} + i_t \sigma_c(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2.15}$$

$$o_t = \sigma_o(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{2.16}$$

$$h_t = o_t \sigma_h(c_t) \tag{2.17}$$

donde σ representa la función de activación para cada compuerta, x_t es la entrada en el instante t para la célula de memoria, W representa a los pesos de la red, así como b representa los sesgos en la red. Además se identifica a i_t, f_t, o_t, c_t como las compuertas de entrada, olvido, salida y activación respectivamente.

Máquinas de soporte vectorial o *SVM* por sus siglas en inglés: Este tipo de aprendizaje es comúnmente empleado para realizar tareas de clasificación. En este algoritmo se busca realizar la separación de los datos de entrada mediante un hiperplano que genera las clases en las cuales se separaran los datos. Con lo que se tiene como objetivo una separación óptima obteniéndose un hiperplano que tenga un margen mayor (máxima distancia con los puntos cercanos al hiperplano) con los puntos mas cercanos a este, en donde el vector formado por los puntos más cercanos al hiperplano son conocidos como vectores de soporte [13].

En este algoritmo se considera la clasificación lineal de los datos $(x_1, y_1), \cdots, (x_n, y_n)$, para el cual y_i obtiene valores de 1 y -1, lo que corresponde a dos clases diferentes en este ejemplo. El hiperplano tiene la forma $f(x) = w^T x + b$, de tal manera que $y_i f(x_i) > 0$ indica que la clasificación es correcta, con los pesos de la red como w y los sesgos en la red como b. Con lo que el hiperplano se puede construir satisfaciendo que $w^T x + b = 0$ para un conjunto de puntos x. Con el termino $\frac{b}{||w||}$ se puede determinar el desplazamiento del hiperplano desde el origen. En el caso de una separación lineal de los datos, se puede definir dos vectores de soporte con las formas $w^T x + b = 1$ y $w^T x + b = -1$, también conocidos como hiperplano negativo e hiperplano positivo, con la distancia entre estos dos vectores definida como $\frac{2}{||w||}$ el cual representa el margen del hiperplano central. En donde se busca optimiza el margen por medio de intentar minimizar a $||w||$.

Si x_i pertenece a la clase 1, se tiene:

$$(w^T x_i + b) \geq 1, \qquad y_i = 1 \tag{2.18}$$

Por otro lado, si x_i pertenece a la clase 2, se tiene:

$$(w^T x(i) + b) \leq -1, \qquad y_i = -1 \tag{2.19}$$

A partir de las desigualdades (2.18) y (2.19) se obtiene el algoritmo:

$$\min_{(w,b)} ||w|| \tag{2.20}$$

$$\text{sujeto a:} \qquad y_i(w^T x_i + b) \geq 1, \qquad i = 1, 2, \cdots, m \tag{2.21}$$

Obteniendo el máximo margen posible en el hiperplano se genera un valor mínimo para el parámetro de los pesos w en la red [13]. En la figura 2.9 se muestra un ejemplo de clasificación aplicando una red de soporte vectorial.

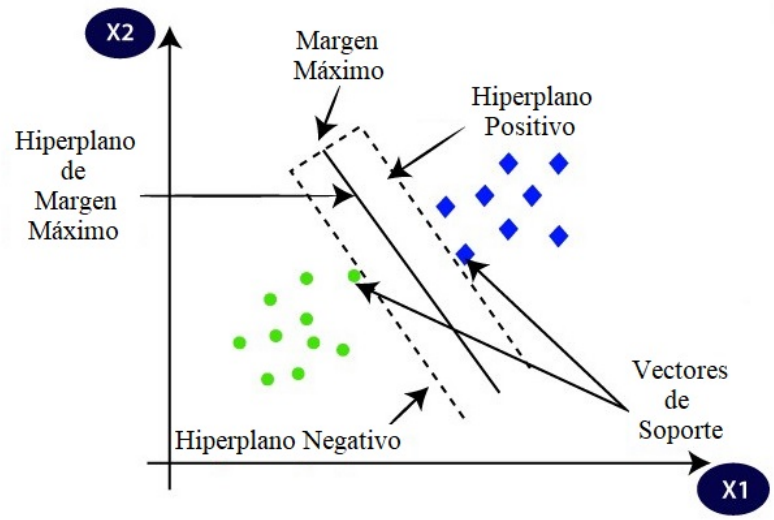


Figura 2.9: Red de vectores de soporte *SVM*.

Redes Neuronales Convolucionales: Estas redes también conocidas como

Convolutional Neural Networks *CNN* por sus siglas en inglés se implementaron principalmente en la identificación de imágenes y su algoritmo está formado principalmente de capas convolución y submuestreo en su estructura. En la capa convolucional como elemento principal de la red se tienen hiper-parámetro que consisten en un conjunto de filtros de convolución $k^{(\ell)}$. Cada filtro realiza una convolución con todo el campo de visión $y^{(\ell-1)}$, produciendo un mapa de características $y^{(\ell)}$ [8]. Cada elemento del mapa de características es como una neurona que tiene relación con la entrada y comparte parámetros con neuronas que se encuentran en el mismo mapa. El $j-$ésimo mapa de características de una capa está dada por:

$$y_j^{(\ell)} = f\left(\sum_{i \in M_j} y_i^{(\ell-1)} k_{ji}^{(\ell)} + b_j\right) \tag{2.22}$$

donde M_j representa el conjunto de entradas seleccionadas y el super índice ℓ representa la capa actual. La convolución transpuesta puede verse como la operación inversa correspondiente a la convolución, también conocida como deconvolución [87]. Mientras que para la capa de submuestreo se busca reducir el tamaño de la entrada en la red, con la posibilidad de tener perdidas de información pero reduciendo los cálculos. La salida de esta capa es de la siguiente forma:

$$y_j^{\ell} = f(\beta_j sub(y_j^{(\ell-1)}) + b_j) \tag{2.23}$$

donde $sub(\cdot)$ representa la función de submuestreo. Usando la operación *max-pool* [62], esta divide la entrada en bloques, y de cada uno de esta conserva el elemento de mayor valor para realizar la reducción del tamaño. En el caso de la reducción también se pueden usar las operaciones de *agrupamiento estocástico* en el cual se elige al azar la activación de la red acuerdo con una distribución multinomial con la cual se asegura que las activaciones no maximales en el mapa de características se puedan utilizar, *agrupamiento* L_p donde se busca

mejorar la generalización de procedimiento de *max-pool*:

$$y_{i,j,k} = \left[\sum_{(m,n)\in\Re_{i,j}} (a_{m,n,k})^p \right]^{\frac{1}{p}} \tag{2.24}$$

con la salida $y_{i,j,k}$ del operados de posición (i, j) en el k-ésimo mapa de características para el valor de posición (m, n) dentro de la región $\Re_{i,j}$ [11] o el *agrupamiento multiescala sin orden* donde se extraen características profundas de activación para la imagen total y para los parches locales a diferentes escalas [64]. Por último este algoritmo cuenta con una capa completamente conectada. Después de las etapas de convolución y submuestreo se cuenta con una configuración de neuronas completamente conectadas y que dictan el número de clasificaciones que se buscan hacer, por lo que esta parte final se utiliza principalmente para realizar clasificaciones usando el operador *softmax*, aunque también se pueden utilizar configuraciones para la identificación de sistemas, lo cual se realiza mediante una sola neurona en la capa de salida y una función de activación como la sigmoide o una función lineal rectificada $ReLU$. Esta ultima se expresa de la siguiente forma:

$$f(x) = \max(0, x) \tag{2.25}$$

Esta función de activación es comúnmente utilizada para estructuras de redes neuronales convencionales o de redes con estructuras profundas debido a que acelera la etapa de aprendizaje. En la figura 2.10 se muestra una representación de las etapas para el algoritmo de una red neuronal convolucional.

Aprendizaje no supervisado.

En el caso del aprendizaje no supervisado se emplean estructuras que por lo general no cuentan con un conocimiento a priori de los datos, por lo que sus procedimientos están basados en observaciones y son tratados como conjunto de datos con variables aleatorias. Este tipo de aprendizajes son aplicados en tareas de agrupaciones, aunque no están limitados

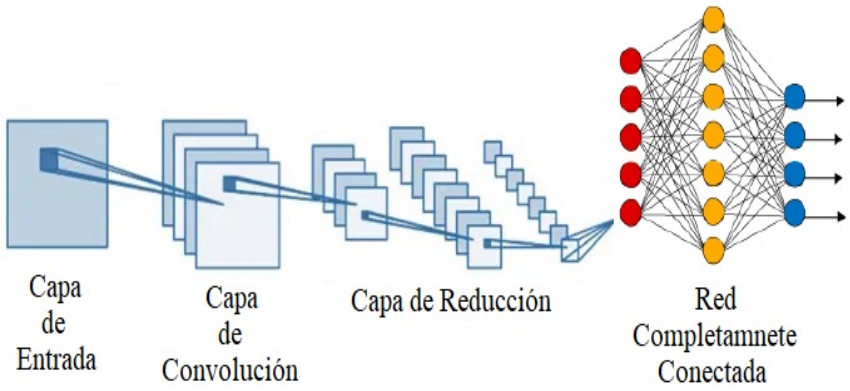


Figura 2.10: Red neuronal convolucional *CNN*.

a esta aplicación. En la figura 2.11 se muestra la estructura con la cual se lleva a cabo el método de aprendizaje no supervisado.

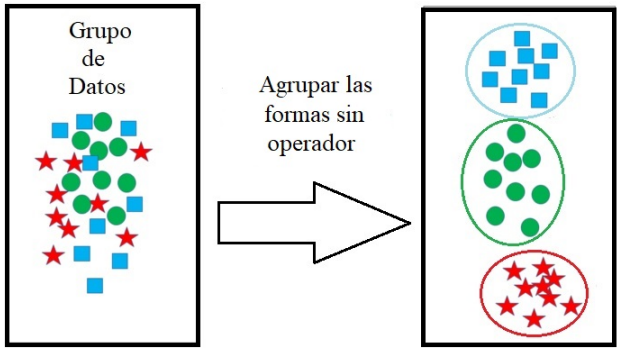


Figura 2.11: Aprendizaje no supervisado.

Aprendizaje por refuerzo.

Este tipo de aprendizaje usa algoritmos que están basados en procesos de realimentación en los cuales se construye un modelo o aprendiz basados en observaciones del mundo que

rodea a la tarea a realizar y otorgando un valor a la acción realizada en cada paso de la tarea. Por lo que este tipo de aprendizaje está basado en las pruebas y análisis del error de cada tarea. En la figura 2.12 se puede observar un esquema de un aprendizaje de tipo reforzado.

Figura 2.12: Aprendizaje por reforzado.

2.2. Teoría Bayesiana

En la vida cotidiana aparecen acontecimientos que nos rodean si saber cómo es que pasaron, pero con base a nuestras experiencias les damos un valor cuantitativo a las características que los rodean para describir la causa que los produce, por ejemplo, cuando realizamos el lanzamiento de una moneda se sabe que se tienen dos posibilidades y asignamos el 50 por ciento a cada una de ellas. A esta asignación de valores se le llama probabilidad. Dentro de las redes neuronales también se pueden aplicar probabilidades Bayesianas en el tipo de aprendizaje que se aplica en estas, con el objetivo de realizar buenas estimaciones de parámetros, predicciones de datos, así como la selección y validación de modelos [29].

2.2.1. Probabilidad de Bayes.

La probabilidad Bayesiana indica una distribución de probabilidad que está condicionada, es decir, los cálculos de una distribucion posterior se llevan a cabo en el contexto de algunas creencias previas, como cuando queremos determinar las condiciones del clima dependiendo

de la estación del año. Pero la asignación de estos valores se dificulta debido a que depende totalmente de la experiencia. Por lo que se puede llevar acabo la asignación de probabilidades a sucesos únicos [3], como lo puede ser el resultado de un enfrentamiento deportivo dados los resultados anteriores de cada equipo. Los juegos comunes en los cuales se presenta la asignación de probabilidades se observan en la figura 2.13

Por otra parte, existe la idea de que la probabilidad se refiere a la frecuencia relativa con que ocurre un suceso, continuando con el ejemplo del lanzamiento de la moneda seria obtener el valor de probabilidad de obtener 2 caras en 5 lanzamientos, con lo que no se podría asignar probabilidades a eventos únicos porque no tendrían frecuencia relativa. Aunque tanto el enfoque clásico de probabilidad como el enfoque de frecuencias relativas cumplen con los axiomas de probabilidad [3].

Figura 2.13: Juegos de probabilidad.

2.2.2. Teorema de Bayes.

Dentro de la teoría de probabilidad se consideran los siguientes axiomas:

1. La probabilidad de un evento se encuentra entre cero y uno.

2. La probabilidad de que el evento suceda con seguridad es igual a uno.

3. La probabilidad de dos eventos independientes es igual a la suma de las probabilidades de cada uno de ellos.

Existe un cuarto axioma que tiene como base la probabilidad condicionada: Si A y B son dos eventos, entonces la probabilidad del evento A, dado que se cumpla B, se representa como $p(A|B)$ [29] y está dada por la siguiente ecuación:

$$p(A|B) = \frac{p(A\ U\ B)}{p(B)} \qquad (cuando\ p(B) \neq 0) \tag{2.26}$$

El teorema de Bayes nos permite calcular la probabilidad condicional a posteriori, la cual es la representación de la probabilidad condicional que existe en los datos (X) basados en una hipótesis (H) resultando en $p(H \mid X)$, con lo cual se obtiene la probabilidad condicionada a partir de los valores de otras hipótesis:

$$p(H \mid X) = \frac{p(H)p(X|H)}{p(X)} \qquad (p(X) \neq 0) \tag{2.27}$$

Este resultado en el enfoque de las probabilidades puede aplicarse al calculo de probabilidad de un suceso único así como al cálculo de probabilidad en frecuencia relativa. Debido al resultado de la fórmula de Bayes se puede observar como una persona cambia sus creencias cuando consigue nueva información [3].

El teorema de Bayes nos permite actualizar las probabilidades previas aplicando la probabilidad de una determinada hipótesis. Por lo que la estimación de probabilidad en una situación similar a la anterior se conoce como inferencia Bayesiana [3].

En resumen, el método de probabilidad de Bayes es una regla para actualizar las creencias sobre una nueva información utilizando los conocimientos previos, por lo cual este método puede ser utilizado como un aprendizaje cuantitativo, teniendo en cuenta de que en casos con muchos datos será difícil formular de manera cuantitativa cuales son las creencias [29].

2.3. Metodología en Redes Neuronales Bayesianas

En las redes neuronales se puede aplicar un enfoque de estadística o probabilidad donde se puede generar la metodología Bayesiana y se conocen como redes neuronales Bayesianas. La diferencia de las redes neuronales Bayesianas con el perceptrón o perceptrón de múltiples capas se encuentra en los resultados ya que el aprendizaje Bayesiano se expresa en términos de una distribución de probabilidad, o función de distribución de probabilidad *probability density function PDF* por sus siglas en inglés. Por lo que el aprendizaje Bayesiano se ejecuta mediante reglas de probabilidad en donde los valores y diseño de la distribución de probabilidad depende totalmente de las creencias relacionadas con las tareas analizadas [52].

2.3.1. Redes Neuronales Bayesianas.

Hay antecedentes y motivaciones en el campo de las redes neuronales Bayesianas, como Rumelhart y McClelland (1986) y los libros de Hertz, Krogh y Palmer (1991), Bishop (1995) y Ripley (1996) quienes introdujeron las teorías del enfoque Bayesiano en las redes neuronales. En las estructuras de redes Bayesianas existen dos clases, una que funciona como estructura de árbol para encontrar probabilidades, y otra que maneja el cálculo de los datos para obtener distribuciones de probabilidad. En la figura 2.14 se observa un ejemplo de red Bayesiana de tipo árbol de probabilidad donde se busca identificar caries en un paciente que siente dolor en su dentadura.

Las Redes Neuronales Bayesianas (*Bayesian neural network* BNN), también son conocidas como redes de creencias o redes de Bayes, estas representan un modelo gráfico probabilístico. Este tipo de estructuras representa el dominio de conocimiento de los eventos de la tarea que se analiza. Los nodos de la red representan a las variables y los enlaces representan sus probabilidades como se observa en la figura 2.14, en donde los enlaces representan dependencias entre las variables aleatorias que se conectan. Estas dependencias son las que se calculan dentro de la red utilizando métodos computacionales y estadísticos.

Con estas características este tipo estructuras para redes neuronales se encuentran

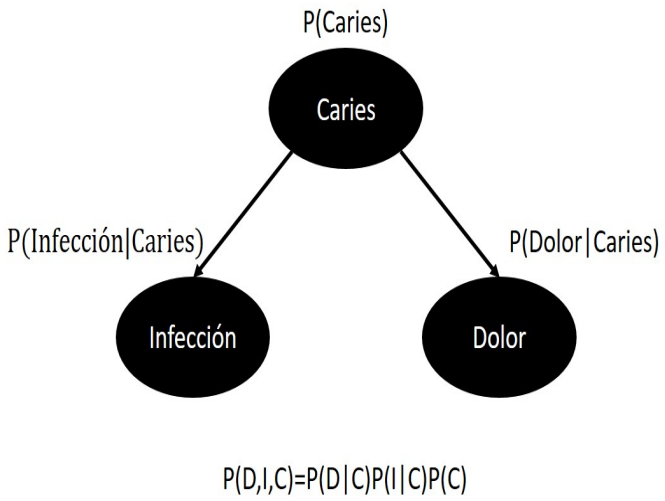


Figura 2.14: Ejemplo de red Bayesiana de tipo árbol.

dentro del campo de análisis de grafos así como dentro de la teoría de probabilidad y estadística.

La estructura gráfica en la red Bayesiana garantiza que no hay ningún nodo que pueda ser su propio antepasado o descendiente. Tal condición es de vital importancia para el cálculo de la probabilidad conjunta de una colección de nodos [4].

Para las estructuras de redes neuronales de tipo árbol de probabilidades se definen los nodos en la gráfica cómo $G = X_1, X_2, ..., X_n$ donde se selecciona un modelo de distribución de probabilidad con los parámetros θ generando los parámetros de la red $B = (G, \theta)$, donde el conjunto de parámetros tiene condicionada la probabilidad π_i para cada una de las acciones x_i en cada uno de los nodos X_i obteniéndose una probabilidad condicional $\theta_{X_i|\pi_i} = P_B(X_i|\pi_i)$. Con lo cual se genera una distribucion de probabilidad para todo el conjunto de nodos en la red:

$$P_B(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P_B(X_i|\pi_i) = \prod_{i=1}^{n} \theta_{X_i|\pi_i} \tag{2.28}$$

En el caso en el que un nodo X_i no tiene un nodo conectado anterior su distribución de probabilidad es local y llamado incondicionado.

2.3.2. Aprendizaje en Redes Bayesianas

El aprendizaje en las redes neuronales está basado en optimizar parámetros de esta por medio de funciones de costo relacionados con el procedimiento, como lo puede ser el error de modelado. En el caso de las redes Bayesianas se busca maximizar la probabilidad del conjunto de datos en los parámetros como parte de su entrenamiento a diferencia de minimizar el error modelado en el perceptrón. Por lo que si se tiene un conjunto de datos de entrenamiento $X = \{x_1, ..., x_m\}$, y un conjunto de parámetros de la distribución de probabilidad como $\Theta = (\theta_1, ..., \theta_n)$ se calcula la máxima verosimilitud L del conjunto de datos de entrenamiento como la suma de cada uno de estos términos, uno para cada nodo:

$$logL(\Theta|X) = \sum_{m} \sum_{n} logP(x_{li}|\pi_i, \theta_i) \tag{2.29}$$

Con lo que cada distribución de probabilidad en los nodos es maximizada independientemente. Pero también se pueden calcular las distribuciones posteriores asignando una distribución previa a cada nodo y actualizándola con los datos de entrenamiento [4].

2.3.3. Estructura general en una red neuronal Bayesiana.

La metodología aplicada al aprendizaje en una red neuronal Bayesiana se puede convertir en un estimado de probabilidades desconocidas. Considérense los siguientes tipos de aprendizaje para una red neuronal de tipo Bayesiana.

1. *El Modelo y función de verosimilitud (likelihood)*: Sea una variable aleatoria independiente $x_i = x_1, ..., x_n$ se puede asignar un modelo probabilístico para calcular las probabilidades en esta variable, en el que un conjunto de parámetros θ determina las distribuciones de probabilidad de x_i Con lo que la distribución de probabilidad condicional de la variable puede ser representada como $P(x_i|\theta)$ x_i. El ejemplo de distribución más utilizado para la variable x_i es una distribución Gaussiana debido al teorema de límite central donde se menciona que las medias de una variable aleatoria tienden a formar una distribución normal o Gaussiana [[29]], donde la distribución normal considera los parámetros de media y desviación estándar como $\theta = (\mu, \sigma)$:

$$P(x_i|\mu, \sigma) = \exp(-(x_i - \mu)^2/2\sigma^2)/\sqrt{2\pi\sigma}. \tag{2.30}$$

Este análisis de la distribución de probabilidad y cálculo de una distribución condicionada se puede ejecutar como aprendizaje en una red neuronal Bayesiana por medio de la función de verosimilitud (likelihood) representada por $L(\theta) = L(\theta|x_1, ..., x_n)$, representado la probabilidad de las observaciones en relación con los parámetros de la distribución de probabilidad. La verosimilitud L es proporcional "$\propto$" a $p(x_i \mid \theta)$ debido a la obtención de valores de la verosimilitud con base a los valores en los parámetros θ [52]:

$$\begin{aligned} L(\theta) &= L(\theta|x_1, ..., x_n) \\ \propto \quad P(x_1, ..., x_n|\theta) &= \textstyle\prod_{i=1}^{n} P(x_i|\theta) \end{aligned} \tag{2.31}$$

Para el ajuste de los parámetros en la red se utiliza la metodología de máxima verosimilitud, en la cual se asignan valores a los parámetros en θ que maximiza la probabilidad en $L(\theta|x_1, ..., x_n)$. La aplicación de estos modelos para el aprendizaje de la red neuronal tiene ventajas como la convergencia al valor verdadero de probabilidad a medida que aumenta la cantidad de datos observables [52].

2. *Aprendizaje y Predicción*: El aprendizaje Bayesiano resulta en una distribución de probabilidad sobre los parámetros del modelo que expresan una creencia sobre los diferentes valores de los parámetros. El primer paso consiste en asignar una probabilidad en los

parámetros $P(\theta)$ que será conocida como información a priori la cual depende totalmente de las creencias para asignarlos.

El segundo paso en el aprendizaje es usar la Regla de Bayes para actualizar la distribución de probabilidad al momento de agregar la información de los datos observados en x_i:

$$\begin{aligned} P(\theta|x_1,...,x_n) &= \tfrac{P(x_1,...,x_n|\theta)P(\theta)}{P(x_1,...,x_n)} \\ &\propto L\left(\theta|x_1,...,x_n\right)P(\theta) \end{aligned} \tag{2.32}$$

La actualización de la distribución de probabilidad en el aprendizaje de la red genera una distribución a posteriori, la cual combina la información de la probabilidad de verosimilitud y la información a priori. En otras palabras, la probabilidad a priori construye una distribución de probabilidad usando la función de verosimilitud.

Finalmente, en este tipo de aprendizaje se puede estimar un valor futuro, o también llamado "predicción" $x_{(n+1)}$, agregando las observaciones del modelo con respecto a la distribución a posteriori de los parámetros:

$$P(x_{(n+1)}|x_1,...,x_n) = \int P(x_{(n+1)}|\theta)P(\theta|x_1,...,x_n)d\theta \tag{2.33}$$

Este proceso de aprendizaje es conocido como inferencia Bayesiana completa para $x_{(n+1)}$ la cual representa una distribución predictiva de $x_{(n+1)}$ dados los datos $x_1,...,x_n$, generando predicciones la cual es la mayor diferencia de una red Bayesiana con el perceptrón [52].

3. *Modelos Jerárquicos*: Cuando el modelo que se utiliza tiene muchos parámetros a determinar $\theta = \{\theta_1,...\theta_p\}$, se puede representar su distribución previa por medio de un hiperparámetro común γ en el cual se contienen todos los parámetros de θ. Estos esquemas se pueden llevar a un gran número de niveles y se conocen como modelos jerárquicos.

Si los parámetros θ_k son independientes dado γ, tendremos la distribución de probabilidad siguiente:

$$P(\theta) = p(\theta_1,...\theta_p) = \int P(\gamma)\prod_{k=1}^{p} P(\theta_k|\gamma)d\gamma \tag{2.34}$$

2.4. Redes neuronales de enfoque probabilístico.

En el estudio de las estructuras de redes neuronales aparecen enfoques probabilísticos como las máquinas restringidas de Boltzman (*Restricted Boltzmann Machine RBM*) que aparecieron en los años ochenta [1], donde se considera la probabilidad $p(x)$ dentro de un vector x para estudiar al sistema con base a creencias y características no observadas en los datos c por medio de la energía de pares booleanos como $E(c,v) = -\sum_i a_i c_i - \sum_j b_j v_j - \sum_i \sum_j c_i \omega_{i,j} v_i$ donde ω representa los pesos y a_i, b_i representa al sesgo de los pesos para la entrada c_i y la salida v_i respectivamente [1]. En la figura 2.15 se observa la estructura de una máquina restringida de Boltzmann:

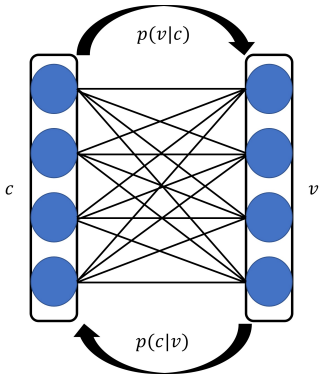


Figura 2.15: Maquina restringida de Boltzmann.

$$p\left(x\right) = \sum_{c} \frac{e^{E(x,c)}}{\sum_{u,g} e^{-e(u,g)}} \tag{2.35}$$

El procedimiento de los cálculos de esta distribución para el aprendizaje de la máquina restringida de Boltzmann están basados en probabilidades condicionales $p(c \mid v)$ y $p(v \mid h)$ definidas por la función de la energía E donde h representa a las características no observadas, describiendo las distribuciones de la siguiente forma:

$$P\left(v_i = y \mid c\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-} \frac{(y - a_i - \sigma_i \sum_j \omega_i j c_j)^2}{2\sigma_i^2} \tag{2.36}$$

$$P(c_j = 1 \mid v) = sigmoide(b_j + \sum_i \frac{v_i}{\sigma_i}\omega_{ij}) \tag{2.37}$$

En las ecuaciones (2.36) y (2.37) los pesos ω_{ij} que relacionan las entradas v_i con las variables en c_j, con los sesgos a_i para para las entradas y b_j para c.

La estructura de las maquinas restringidas de Boltzmann se puede modificar conectando en serie estructuras de la máquina restringida de Boltmann simple, donde solo la primera capa tiene como entrada los valores reales del sistema analizado y las otras capas son estructuras binarias de probabilidad, siendo $p(c_1 \mid v)$ la nueva entrada a las segunda celda de la máquina restringida de Boltzmann para poder calcular la probabilidad condicional de $p(c_2 \mid c_1)$, esta estructura se puede observar en la figura 2.16.

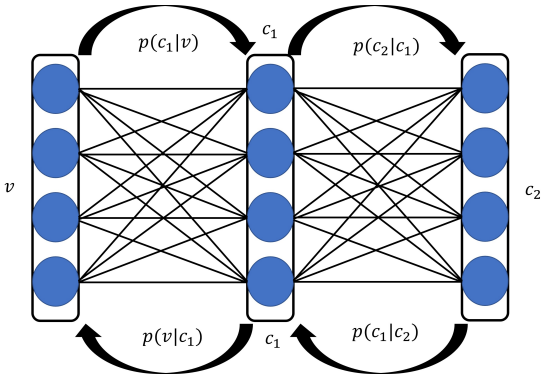


Figura 2.16: Estructura profunda para una maquina restricta de Boltzman.

2.5. Identificación de sistemas dinámicos con aprendizaje automático.

Es posible generar un modelo mediante el análisis de fenómenos físicos y químicos que se representan mediante conceptos matemáticos y que indican el funcionamiento de un sistema. Para la gran mayoría de los sistemas nos encontramos con representaciones no lineales, pero

en algunos casos se pueden generar representaciones de modelo lineal debido a que estos también pueden representar un caso práctico además de ser fáciles de manipular [2]. Estos modelos permiten generar simulaciones de la dinámica obtenida además permite diseñar un control para manipular al sistema.

Aunque en principio generar un modelo suena fácil, cuando se opta por el empleo de fenómenos físicos o químicos se debe tener un conocimiento suficientemente del sistema para determinar la mayoría de las dinámicas que lo rigen y generar valores o ajustes de sus parámetros.

El objetivo en la identificación de los sistemas dinámicos por medio de redes neuronales es generar un modelo que pueda describir al sistema, cuando no se tiene un conocimiento suficiente del mismo. En lo general cuando no se conoce el detalle de un modelo del sistema, pero se tiene disponible los datos de entrada y salida estos se representan como series de tiempo. Los datos de entrada y salida del sistema pueden generar información para crear un modelo de la planta a identificar, donde generalmente el sistema que se genera puede representarse por medio de ecuaciones diferenciales [36].

En el aprendizaje automático nos encontramos con la opción de las redes neuronales para modelar un sistema. Las redes neuronales son modelos de caja negra o caja gris debido a que no necesita contar con la información completa del sistema a modelar, y es posible generara un modelo sólo con analizar los datos que se tienen en la entrada y salida del sistema. En la ejecución de estos modelos basados sólo en los datos disponibles se tienen algunas desventajas como lo es el diseño de la estructura debido a que no existe un método que permita generar los parámetros con los que cuentan las redes neuronales entre los que se encuentran la cantidad de capas en la red neuronal, así como el número de los nodos en cada una de las capas [41]. Por otro lado, una red neuronal puede modelar un sistema dinámico seleccionando una estructura con una capa e ir aumentando el número de nodos, como lo dicta el teorema de aproximación universal de redes neuronales [31]. Mientras que si se seleccionan un número grande de nodos para la red neuronal de una capa este tiende a generar un sobre ajuste en la red [70].

2.5.1. Identificación de sistemas con enfoques Bayesianos.

Existen procesos similares al aprendizaje automático enfocados en las teorías Bayesianas para la identificación de los sistemas dinámicos. Principalmente nos encontramos a los procesos gaussianos (*Gaussian Process GP*) para generar modelos que identifican sistemas dinámicos. Los procesos Gaussianos se basan en obtener el modelo de un sistema generando distribuciones de probabilidad, estos modelos generan sus propios parámetros, como la media y varianza en una distribución normal, y realizan el ajuste de estos parámetros mediante la maximización de la verosimilitud en las distribuciones.

Estos modelos también presentan problemas. Primero, sin la experiencia acerca del modelo a interpretar es posible especificar mal la función del modelo más adecuada. En segundo lugar, los procesos gaussianos pueden ser limitados y no ser capaces de capturar algunos tipos de comportamientos no lineales. Por esta razón este tipo de modelos son empleados para modelar series de tiempo [6], [34], [26], [37], y [54].

Este tipo de modelos también está basado en los datos, con las características de representar distribuciones utilizando a estos como variables aleatorias y estas forman un modelo de probabilidad [84]:

$$\begin{aligned} p(y|X,w) &= \prod_{i=1}^{n} p(y_i|x_i,w) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{\left(y_i - x_i^T w\right)^2}{2\sigma_n^2}\right) \\ &= \frac{1}{(2\pi\sigma_n^2)} \exp\left(-\frac{1}{2\sigma_n^2}\left|y - X^T w\right|^2\right) \\ &= N(X^T w, \sigma_n^2 I) \end{aligned} \tag{2.38}$$

Este tipo de modelado genera una predicción de los datos por medio del cálculo de una distribución a posteriori, en la cual se tiene un conocimiento a priori de las distribuciones que está totalmente basado en los conocimientos y observaciones del usuario de una distribucion normal N. Con lo que el modelado del sistema por medio de los procesos Gaussianos GP

queda en función de los parámetros en el proceso gaussiano.

$$f(x) \sim GP\left(m(x), k(x, x')\right) \tag{2.39}$$

En el proceso se generan parámetros que provienen directamente del modelo de distribución de probabilidad como la función de la media $m(x)$ y la función de covarianza $k(x, x')$. Con el manejo de estos parámetros para el proceso Gaussiano se busca optimizar el valor de la verosimilitud del proceso:

$$\theta_{opt} = \arg\max_{\theta} p\left(y|X, \theta\right) \tag{2.40}$$

En la figura 2.17 se representa un ejemplo de identificación de una serie de tiempo por medio de un proceso Gaussiano.

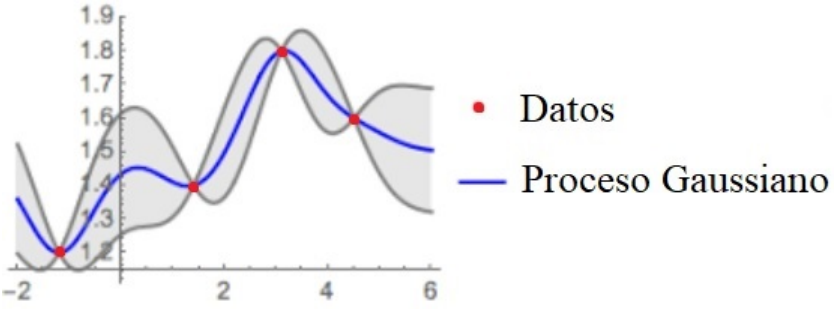


Figura 2.17: Proceso gaussiano para identificación de sistemas dinámicos.

2.5.2. Aprendizaje profundo en Redes de tipo Bayesianas

La combinación que surge de la estructura de una red neuronal profunda y el proceso estocástico de una red Bayesiana se denomina aprendizaje profundo Bayesiano (*BDL Bayesian Deep Learning*) [88].

Estas arquitecturas pueden modelar tareas complejas aprovechando el poder de representación jerárquica del aprendizaje profundo, a la vez que pueden inferir distribuciones

posteriores. Los modelos Bayesianos de aprendizaje profundo típicamente forman estimaciones de incertidumbre, ya sea empleando distribuciones para definir a los pesos del modelo, o aprendiendo un modelo de distribucion de probabilidad [35]. Recientemente el aprendizaje profundo Bayesiano está experimentado un resurgimiento de interés en las investigaciones, esto puede ser ocasionado debido a que los sistemas automáticos con aprendizaje profundo enfocados en estimaciones de incertidumbre obtienen buenos resultados [53].

Para calcular progresivamente en cada capa de la red los pesos, la unión de las técnicas de aprendizaje automático principalmente consisten en la aplicación de un regresor lineal de tipo Bayesiano a la última capa de una red neuronal profunda, restringiendo sólo los pesos de la capa final a un proceso Bayesiano y calculando de forma convencional los parámetros restantes. De esta manera el tiempo de evaluación y de modelo se reduce [65].

Estructuras generales de aprendizaje profundo Bayesiano.

En este caso se busca unificar la teoría básica del aprendizaje profundo con el enfoque Bayesiano en el funcionamiento de una red neuronal Bayesiana, recordando que una de las grandes ventajas de este tipo de redes neuronales es trabajar con un conjunto de datos que puede contener los siguientes tipos de incertidumbre [81]:

1. Incertidumbre en los parámetros de la red neuronal.
2. Incertidumbre en los parámetros específicos de la tarea.
3. Incertidumbre de intercambio de información entre componente de percepción y el componente específico de la tarea.

2.5.3. Enfoque Bayesiano para control de sistemas dinámicos.

El aprendizaje profundo Bayesiano también se puede aplicar al control de sistemas dinámicos no lineales. Si se desea controlar un sistema dinámico complejo, una forma de resolver este problema de control es mediante la iteración entre dos tareas, la tarea de

identificación y la tarea de control. La etapa de identificación se puede calcular usando múltiples capas de simple transformación no lineal (aprendizaje profundo) mientras que en la etapa de control se aplica un modelo de control al sistema. Se necesita de una comunicación entre los procesos de modelado de datos y control para que la ejecución sea óptima. Siendo el proceso de modelado de donde se obtienen los parámetros para realizar la acción de control [81].

Capítulo 3

Enfoque Bayesiano para pronóstico de series de tiempo

La inferencia Bayesiana es una de las herramientas más útiles en la predicción de eventos, con la característica de representar distribuciones de probabilidad futuras de esos eventos, con lo cual nos acercamos a resultados de predicción de eventos importantes en la vida cotidiana como lo es la estimación de eventos meteorológicos o clima, estimación de eventos sísmicos, estimación de valores de mercado de bolsa o el caso más utilizado que es la estimación de resultados en eventos o competencias deportivas.

En este trabajo hacemos un énfasis en la estimación de parámetros sísmicos en regiones de alta actividad sísmica dada su importancia y los riesgos que involucran estos eventos para las personas. Y nos encontramos con un auge en la identificación de estos eventos gracias a una variedad de herramientas implementadas. Las teorías Bayesianas han tenido un gran crecimiento en su aplicación por la facilidad de su implementación, empezando con aplicaciones de los años ochenta y noventa en las zonas de Grecia [12], [66], [67] para predicción de sismos fuertes en su magnitud, así como trabajos recientes, Wang, et. al. en el 2015, con algoritmos Bayesianos para predecir sismos próximos en Taiwán [82]. Un trabajo desarrollado para predicción de sismos fuertes en su magnitud Wang, et. al. en el 2015, así como la aplicación de algoritmos Bayesianos para predecir sismos en un región de Taiwán

[48].

Debido a su aplicación en predicción, los procedimientos Bayesianos surgen como aplicación de alertas tempranas a estos acontecimientos importantes, además de que estas metodologías pueden ser implementadas cuando los datos son imprecisos o surgen pérdidas en los datos [16],[14].

3.1. Modelos de distribución para Inferencia Bayesiana.

El enfoque Bayesiano para producir distribuciones de probabilidad está basado en la probabilidad condicional donde se cumplen con todos los axiomas de la probabilidad básica como se mostró en el capítulo 2 de este trabajo. Los resultados de las distribuciones de probabilidad generados por medio del teorema de Bayes representan información de eventos futuros.

Con base en la teoría de Bayes se pueden determinar las distribuciones posteriores o a posteriori de datos que se podrían representar como series de tiempo, las cuales representa la información que conocemos y los datos observables del sistema, y que en el enfoque de la probabilidad este también puede ser visto como una variable aleatoria:

$$x = (x_1, x_2, ..., x_n) \tag{3.1}$$

Ya que se ha conocido la información a priori del sistema analizado se tienen el enfoque de probabilidad donde si se conocen dos eventos A y B se pueden generar las probabilidades de dichos eventos como $p(A)$ y $p(B)$ además de una probabilidad condicional del evento B dado el evento en A como $p(B \mid A)$, o en el caso de una probabilidad condicionada de un evento A dado que ocurrió el evento en B como $p(A \mid B)$. Con lo que se puede aplicar el teorema de Bayes para obtener una distribución de probabilidad que pertenecerá eventos posteriores:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{3.2}$$

La distribución a posteriori puede ser calculada mediante eventos aleatorios, donde toda la evidencia de los eventos es tomada en cuenta en la cual se puede seleccionar el modelo a seguir en la distribución de probabilidad, por lo que la distribución posterior puede ser representada de la siguiente forma:

$$p(\theta|x) \propto p(\theta)p(x|\theta) \tag{3.3}$$

En el teorema de Bayes en la ecuación (3.3) $p(\theta)$ representa la distribución de probabilidad con un conocimiento a priori, y dada una variable observada x se tienen una probabilidad de verosimilitud $p(x|\theta)$.

En los métodos de Monte Carlo se pueden generar distribuciones de probabilidad, así como calcular predicciones en la distribución con base a los parámetros del modelo de distribución que se selecciona facilitando el cálculo de una distribución a posteriori ya que no se calculan probabilidades condicionales de los parámetros θ dados los datos de análisis x [29]. Debido a que nos hemos encontrado con distribuciones normales y exponenciales en los sismos analizados en el capítulo 5.1. de este trabajo se analiza el modelo de distribución normal y exponencial así como el método de Monte Carlo para estas distribuciones y generar un procedimiento para el cálculo de distribuciones posteriores que se aplique a los parámetros sísmicos en Italia.

3.1.1. Modelos de distribución.

Para un conjunto de variables se pueden seleccionar modelos con los cuales calcular sus probabilidades, aunque en la teoría tenemos el teorema de límite central que dicta que la suma de todas las medias en las distribuciones tienden a una distribución Gaussiana (o también llamada distribución normal) [29].

Modelo Gaussiano o normal.

El modelo de distribucion normal también es conocido como modelo de distribucion Gaussiana o campana de Gauss por la forma que obtiene. En la figura 3.1 se observa un ejemplo de la distribución de tipo normal para una variable aleatoria con media cero.

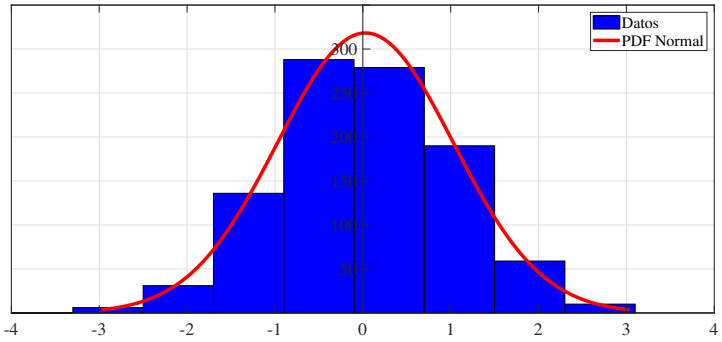


Figura 3.1: Distribución con modelo normal.

Para la construcción de una distribución normal en una variable aleatoria x se depende completamente de los parámetros estadísticos promedio μ y desviación estándar σ:

$$p\left(\mu,\sigma^2|x\right)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]=P\left(\mu,\sigma^2\right) \tag{3.4}$$

El modelo de distribución para una variable x también puede denotarse de la siguiente forma:

$$x\sim N\left(\mu,\sigma\right) \tag{3.5}$$

Dentro del enfoque estadístico la distribución normal cumple con las siguientes propiedades:

1) La moda de la distribución normal es única y es equivalente a su media y mediana.

2) El área bajo la curva de la distribución es igual a 1.

3) La curva de la distribución es simétrica con respecto de la media μ.

4) El área bajo la curva entre el intervalo $(\mu - 1{,}96\sigma, \mu + 1{,}96\sigma)$ es igual a 0.95.

En el caso que una variable x tenga una distribución normal $N(\mu, \sigma)$ se puede calcular la probabilidad de un valor con respecto de los parámetros de media μ y desviación estándar σ.

$$Z = \frac{x - \mu}{\sigma} \tag{3.6}$$

Para la distribución normal se tiene la media muestral con respecto del número de muestras n como se presenta a continuación:

$$\bar{x} \sim N\left(\mu, \frac{\sigma}{\sqrt{n}}\right) \tag{3.7}$$

El resultado de la ecuación (3.7) se debe al teorema de límite central el cual dicta que las medias en las muestras aleatorias independientes e idénticamente distribuidas de la variable siguen una distribución normal con la media de la población y desviación estándar de la población entre $\sqrt{n}$. Este mismo resultado puede representarse en el intervalo $\left(\mu - \frac{1{,}96\sigma}{\sqrt{n}}, \mu + \frac{1{,}96\sigma}{\sqrt{n}}\right)$ cumpliendo con la propiedad 4 de la distribución normal para el cual se conocerán el 95 % de los valores de la media muestral.

En el caso cuando se presentan dos distribuciones de tipo normal X y Yse puede calcular la diferencia entre las distribuciones por medio de la divergencia de *Kullback-Leibler*:

$$DKL(X \| Y) = \frac{1}{2}\left(\log\left(\frac{\sigma_Y^2}{\sigma_x^2}\right) + \frac{\sigma_Y^2}{\sigma_x^2} + \frac{(\mu_Y - \mu_x)^2}{\sigma_Y^2} - 1\right) \tag{3.8}$$

3.1.2. Monte Carlo para modelar distribuciones normales.

El método de Monte Carlo se utiliza para obtener las distribuciones de probabilidad. Donde se emplean muestras de $x_1 \cdots x_n$ para obtener la distribución $p(x \mid \theta)$ donde θ

representa a los parámetros del modelo de distribución. Modelando la distribución de probabilidad $p(x)$:

$$p\left(x_i\right) \approx \frac{x_i}{\sum_{i=1}^{n} x_i} \tag{3.9}$$

Una de las ventajas de la aplicación del método de muestreo de Monte Carlo es que si se tiene un número grande de observaciones la aproximación de la densidad es alta [29]. Por lo que aproximamos a los datos como $\hat{x}$ dados los parámetros en θ, es decir, obtenemos resultados para $p\left(x = \hat{x} \mid x_1 \cdots x_n\right)$, y se puede obtener la distribución posterior $p\left(\theta \mid x\right)$:

$$p\left(\theta \mid x\right) = \int p\left(\hat{x} \mid \theta\right) \hat{p}\left(\theta \mid x_1 \cdots x_n\right) d\theta \tag{3.10}$$

En la ecuación (3.10) $\hat{p}\left(\theta \mid x_1 \cdots x_n\right)$ representa la distribución de las muestras en el procedimiento de Monte Carlo mientras que $p\left(\hat{x} \mid \theta\right)$ representa la expectativa de la distribución posterior.

Realizar la metodología de Monte Carlo para generar una distribución a posteriori se convierte en una modelación por medio de la estimación de los parámetros del modelo de distribución como lo es $\theta = [\mu, \sigma^2]$ en el modelo normal. Realizaremos el caso para el cual se fija la varianza σ^2 y se trabaja con la media μ en el procedimiento. Para el conjunto de datos x con n número de datos observados como $\{x_1 \ldots x_n\}$ con una distribución normal $\left(x_1 \ldots x_n \mid \mu, \sigma^2\right) \sim N\left(\mu, \sigma^2\right)$. La distribución a posteriori está relacionado con el conocimiento en las parámetros a priori de media cuando esta tiene una distribucion normal $\mu \sim N(\mu_0, \tau_0^2)$ obteniendo un modelo para la media de la distribucion posterior μ_n y una varianza para la distribucion posterior τ_n^2:

$$\mu_n = \frac{\frac{1}{\tau_0^2}\mu_0 + \frac{n}{\sigma^2}\bar{x}}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}} \tag{3.11}$$

$$\tau_n^2 = \frac{1}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}} \tag{3.12}$$

En el caso en el que la desviación σ^2 es calculada se usa la familia de la distribución de tipo Gamma como:

$$\begin{aligned} \tfrac{1}{\sigma^2} &\sim \gamma\,(a,b) \\ \sigma^2 &\sim \gamma^{-1}\,(a,b) \end{aligned} \tag{3.13}$$

Definiendo los datos para $a = \frac{v_0}{2}$ y $b = \frac{v_0}{2}\sigma^2$ con σ_0^2 como la varianza y v_0 como el número de muestras de las observaciones en la información a priori, resultando la distribución a posteriori de la siguiente forma:

$$p\left(\mu, \sigma^2 \mid x_1 \ldots x_n\right) = p\left(\mu \mid \sigma^2, x_1 \ldots x_n\right) p\left(\sigma^2 \mid x_1 \ldots x_n\right) \tag{3.14}$$

El procedimiento de Monte Carlo para generar la distribución a posteriori se puede obtener de un número de muestras k_0 para las observaciones en la cual se parametriza a la varianza por medio de una distribucion Gamma $1/\sigma^2 \sim \gamma\left(\frac{v_0}{2}, \frac{v_0}{2}\sigma_0^2\right)$ obteniendo una distribucion para la media como $(\mu \mid x, \sigma^2) \sim N(\mu_n, \sigma/k_n)$ con $k_n = k_0 + n$:

$$\left(\mu \mid x_1 \ldots x_n, \sigma^2\right) \sim N\left(\mu_n, \frac{\sigma^2}{\kappa_n}\right) \tag{3.15}$$

Estos métodos de muestreo de los parámetros de la distribución se pueden incluir en el cálculo de distribuciones condicionales para obtener la distribución de probabilidad a posteriori con respecto de los parámetros del modelo normal cuando se fija a la variable de varianza en toda la población σ^2:

$$\begin{aligned} &p\left(\sigma^2 \mid x_1 \ldots x_n\right) \propto p\left(x_1 \ldots x_n \mid \sigma^2\right) p\left(\sigma^2\right) \\ &= p\left(\sigma^2\right) \int p\left(x_1 \ldots x_n \mid \sigma^2, \mu\right) p\left(\mu \mid \sigma^2\right) d\mu \end{aligned} \tag{3.16}$$

Resultando que el parámetro de la varianza este representado como:

$$\left\{\frac{1}{\sigma^2} \mid x_1 \ldots x_n\right\} \sim \gamma\left(\frac{\upsilon_n}{2}, \frac{\upsilon_n}{2}\sigma_n^2\right) \tag{3.17}$$

donde $\upsilon_n = \upsilon_0 + n$, $\sigma_n^2 = \frac{1}{\upsilon_n}\left(\upsilon_0\sigma_0^2 + (n-1)\,s^2 + \frac{\kappa_0 n(\bar{x}-\mu_0)^2}{\kappa_n}\right)$, $s^2 = \frac{1}{n-1}\sum(\bar{x}-\mu_0)^2$ son las muestras de la varianza. Obteniendo el modelo de la distribución posterior para los parámetros μ dados los datos $(x_1 \ldots x_n)$ en la siguiente ecuación:

$$\sigma^2 \sim \gamma^{-1}(a,b), \quad \left\{\theta \mid x_1 \ldots x_n, \sigma^2\right\} \sim N\left(\mu_n, \tau_n^2\right) \tag{3.18}$$

Con lo que se puede generar una predicción observada en los datos como $\hat{x}$ basada en los datos observados $\{x_1 \ldots x_n\}$, con la distribucion $\{\hat{x} \mid \mu, \sigma^2\} \sim N(\mu, \sigma^2)$ ademas de que se tiene una desviación en el modelo ε como $\hat{x} = \mu + \varepsilon$, la desviación ε tiene una distribución normal $\{\varepsilon \mid \mu, \sigma^2\} \sim N(0, \sigma^2)$ o en otras palabras el modelo de predicción presenta una desviación con una perturbación de distribución normal con medio cero:

$$(\hat{x} \mid \sigma^2, x_1, ..., x_n) \sim N(\mu_n, \tau_n + \sigma^2) \tag{3.19}$$

El método de Monte Carlo para los parámetros del modelo normal tiene los siguientes pasos:

1. Generar muestras para la desviación estándar como $\sigma \sim \gamma^{-1}\left(\frac{\upsilon_n}{2}, \frac{\upsilon_n}{2}\sigma_n^2\right)$.

2. Generar muestras para la media $\mu \sim N\left(\mu_n, \sigma^{2(1)}/k_n\right)$.

Con lo que se obtiene el estimador de la media μ en la media a posteriori por medio de la esperanza:

$$\begin{aligned} \hat{\mu}_b &= E\left[\mu \mid x_1 \ldots x_n\right] \\ \hat{\mu}_b &= \mu_n = \tfrac{n}{k_0+n}\bar{x} + \tfrac{k_0}{k_0+n}\mu_0 = \omega\bar{x} + [1-\omega]\,\mu_0 \end{aligned} \tag{3.20}$$

Con $\omega = \frac{n}{k_0+n}$. Además en el comportamiento del análisis en la ecuación (3.20) se puede obtener dos casos del estimador de la media con el verdadero valor de la media en la población μ^*:

$$\begin{aligned} &\text{Caso 1} \qquad E\left[\hat{\mu}_e \mid \mu = \mu^*\right] = \mu^*, \\ &\text{Caso 1} \qquad E\left[\hat{\mu}_b \mid \mu = \mu^*\right] = \omega\mu^* + \left[1-\omega\right]\mu_0 \end{aligned} \tag{3.21}$$

En el caso 2 de la ecuación (3.21) si $\mu_0 \neq \mu^*$ se dice que el estimador de la media $\hat{\mu}_b$ tiene sesgo, mientras que $\hat{\mu}_e$ representa a un estimador sin sesgo.

Análisis del sesgo en el procedimiento de Monte Carlo para el modelo normal.

Con el sesgo se conoce qué tanto se aleja la estimación del modelo de la probabilidad real. Con lo que se analiza que tanto el estimador de la media $\hat{\mu}$ se acerca a la media objetivo μ^*. Por lo que se puede analizar esta desviación por medio del error medio cuadrático (MSE) en la estimación de la distribución por medio de la varianza VAR en la media μ como $E[(\hat{\mu}-m^2) \mid \mu^*]$ y el sesgo en el modelo $Sesgo$ para la media modelada como $E[(m-\mu^*)^2 \mid \mu^*]$ con $m = E[\hat{\mu} \mid \mu^{ast}]$:

$$MSE\left[\hat{\mu} \mid \mu^*\right] = Var\left[\hat{\mu} \mid \mu^*\right] + Sesgo^2\left[\hat{\mu} \mid \mu^*\right] \tag{3.22}$$

Modelo exponencial

Se tiene una función de distribución de probabilidad (PDF) para un modelo exponencial debido a que con mucha frecuencia nos encontramos este tipo de modelos en los datos analizados. La aplicación de un modelo de distribución exponencial suma resultados al análisis de distribuciones de probabilidad sin la necesidad de forzar el cálculo de las distribuciones a un modelo Gaussiano. La distribución del modelo exponencial para una variable aleatoria x y que se define con esta distribución $x \sim exponencial(\lambda)$ queda representado como:

$$p(x \mid \lambda) = \lambda e^{-\lambda x} \tag{3.23}$$

En el caso del modelo exponencial se tiene un único parámetro λ del cual depende la distribución y este representa la media en la distribución. Este modelo puede obtener

la probabilidad acumulada para un punto de información a analizar dentro de los datos mediante la función de probabilidad acumulada (CDF) del modelo exponencial:

$$p(x \mid \lambda) = 1 - e^{-\lambda x} \tag{3.24}$$

La distribución de probabilidad exponencial cumple con las propiedades de media con respecto de la esperanza E de la distribucion y su varianza, las cuales se pueden seguir con respecto del valor λ de la distribución exponencial:

Media:

$$E[x] = 1/\lambda \tag{3.25}$$

Varianza:

$$Var[x] = 1/\lambda^2 \tag{3.26}$$

Mientras que si se conoce la distribución de probabilidad exponencial y queremos determinar al parámetro λ se puede generar una función de verosimilitud de este parámetro para un conjunto de datos $x = x_1, ..., x_n$:

$$L(\lambda) = \prod_{i=1}^{n} \lambda exp(-\lambda x_i) = \lambda^n \left(-\lambda \sum_{i=1}^{n} x_i \right) = \lambda^n exp(-\lambda n \bar{x}) \tag{3.27}$$

Para la distribución exponencial como en la ecuación (3.27) se tiene que la media con respecto de la muestra de los datos es $\bar{x} = \frac{1}{2} \sum_{i=1}^{n} x_i$. Con lo que se puede definir la derivada del logaritmo natural de la función de verosimilitud de la siguiente forma:

$$\frac{d}{d\lambda} \ln L(\lambda) = \frac{d}{d\lambda} (n \ln(\lambda) - \lambda n \bar{x}) = \frac{n}{\lambda} - n\bar{x} \tag{3.28}$$

La derivada en la ecuación (3.28) queda definida en función del valor de λ:

$$\frac{n}{\lambda} - n\bar{x} \begin{cases} > 0 & si \quad 0 < \lambda < 1/\bar{x} \\ = 0 & si \quad \lambda = 1/\bar{x} \\ < 0 & si \quad \lambda > 1/\bar{x} \end{cases} \tag{3.29}$$

Con lo que se obtiene el valor de la máxima verosimilitud para el parámetro λ de la distribución exponencial:

$$\widehat{\lambda} = \frac{1}{\bar{x}} \tag{3.30}$$

Además, la distribución exponencial puede generar una distribución predictiva generando la forma inferencia Bayesiana con el modelo exponencial. En donde se denota que la distribución exponencial representa un caso especial de la distribución Gamma G con lo cual se pueden desarrollar los cálculos de la distribución a posteriori:

$$\begin{array}{l} \Gamma(\lambda \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda^{\alpha-1} exp(-\lambda\beta) \\ \text{con} \qquad \Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} dt \end{array} \tag{3.31}$$

El parámetro $\alpha > 0$ sitúa la máxima probabilidad de la distribución conociéndose como centro de la distribución y cuando este valor se acerca a cero la distribución toma forma de una distribución exponencial, mientras que el parámetro β representa al tiempo promedio el cual define la simetría de la forma de distribución Gamma obteniendo la razón de cambio como $\lambda = 1/\beta$. La distribución a posteriori del modelo exponencial puede generarse utilizando una previa de modelo Gamma:

$$p(\lambda) \propto L(\lambda) \times G(\lambda \mid \alpha, \beta) \tag{3.32}$$

Por lo que la distribución posterior del modelo exponencial se genera con respecto del parámetro de velocidad λ mediante una distribución Gamma:

$$\begin{array}{rcl} p(\lambda) & = & \lambda^n \exp\left(-\lambda n \bar{x}\right) \times \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp\left(-\lambda\beta\right) \\ p\left(\lambda\right) & \propto & \lambda^{(\alpha+n)-1} \exp\left(-\lambda\left(\beta + n\bar{x}\right)\right) \end{array} \tag{3.33}$$

donde la distribucion posterior puede quedar representada en términos de la distribucion Gamma:

$$p(\lambda) = \Gamma(\lambda \mid \alpha + n, \beta + n\bar{x}) \tag{3.34}$$

Finalmente dadas dos distribuciones con un modelo exponencial se puede calcular una divergencia o distancia que se tiene entre estas mediante la divergencia de Kullback-Liebler (KL):

$$KL(\lambda \parallel \lambda_0) = log(\lambda) - log(\lambda_0) + \frac{\lambda_0}{\lambda} - 1 \tag{3.35}$$

donde λ_0 representa la media en la distribución aproximada $exp(\lambda_0)$, mientras que λ es la media de la distribución exponencial verdadera. En la figura 3.2 se representa un ejemplo de la distribución exponencial para una variable aleatoria.

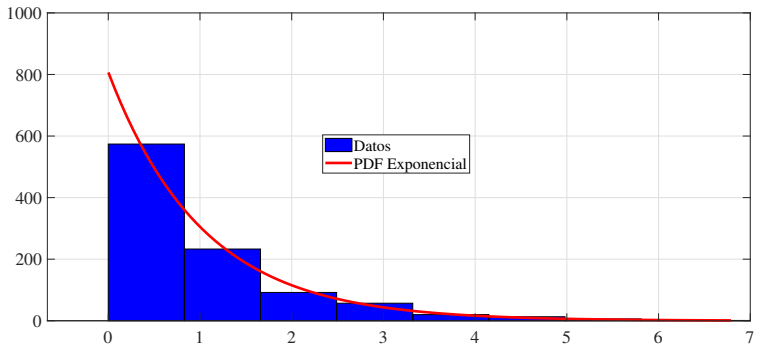


Figura 3.2: Distribución empleando modelo exponencial.

3.1.3. Método de Monte Carlo para obtener una distribucion a posteriori del modelo exponencial.

Se puede emplear el método Monte Carlo para generar el modelo de distribuciones de probabilidad a posteriori como herremainta de inferencia Bayesiana, una opción presentada en lugar del teorema de Bayes.

Para generar las distribuciones en un modelo de distribución exponencial se genera el parámetro de interés θ el cual representa a los parámetros del modelo exponencial λ, aunque se debe tomar en cuenta que la distribución exponencial se calcula mediante el modelo Gamma o el modelo de Poisson con la distribucion para una variable x $p(x) = e^{-\lambda}\lambda^n/n!$. Sea $x = x_1, ..., x_n$, se desea generar la muestra de distribución $p(x \mid \theta)$ con S número de

muestras independientes para los valores de θ en la distribución posterior considerando los siguientes pasos:

1. Generar muestras para los parámetros de la distribución $\theta^{(1)}, \cdots \theta^{(S)} \sim$ i.i.d. $p(\theta \mid x, \cdots x_n)$.

2. Aproximar la distribución $p(\hat{x} \mid x_1 \cdots x_n)$ por medio de $\frac{1}{S}\sum_{i=1}^{S} p\left(x \mid \theta^{(i)}\right)$.

donde $\hat{x}$ representa las observaciones de los datos a predecir y $\frac{1}{S}\sum_{i=1}^{S} p\left(\hat{x} \mid \theta^{(i)}\right)$ necesita de la distribución $p(x \mid \theta)$ lo cual se puede obtener directamente de los datos [29].

Las muestras del modelo predictivo $\hat{x}$ se pueden obtener en los siguientes pasos:

1. Generar muestras para $\theta^{(i)} \sim p(\theta \mid x, \cdots x_n)$.

2. Generar muestras de $\hat{x}^{(i)} \sim p\left(\hat{x} \mid \theta^{(i)}\right)$.

En la secuencia $\left(\hat{x}^{(i)}, \theta^{(i)}\right)$ se tienen S número de muestras independientes para la distribución posterior conjunta de $(\theta, \hat{x})$. Mientras que la secuencia en $\hat{x}^{(i)}$ contiene S muestras independientes para la distribución posterior marginal de $\hat{x}$, la cual representa a la distribución predictiva. En el caso del modelo de Poisson en $p(\hat{x} \mid \theta)$ se genera un modelo de distribucion Gamma para $p(\theta \mid x_1 \cdots x_n)$ resultando la distribución posterior predictiva es un caso negativo de la distribución como $Gamma(\alpha + \sum x_i, \beta + n)$:

1. Generar muestras $\theta^{(i)} \sim \Gamma\left(\alpha + \sum x^{(i)}, \beta + n\right)$.

2. Generar muestras de $\hat{x}^{(1)} \sim Poisson\left(\theta^{(i)}\right)$.

3.2. Modelado de la Probabilidad a posteriori de los modelos normal y exponencial.

Una de las ventajas de conocer los datos a analizar es que se puede obtener el modelo de distribución que se tiene en estos, con lo que se pueden seleccionar el tipo de modelo de distribución para que el modelo se asemeje a la distribución verdadera y con esto implementar el cálculo de distribuciones a posteriori con el modelo previamente seleccionado. En este trabajo se implementó el método de Monte Carlo como inferencia Bayesiana seleccionado el modelo de distribución normal o exponencial, para el cálculo de las distribuciones a posteriori de parámetros sísmicos como se puede observar de los resultados obtenidos en la sección 5.1 de este trabajo. El método de Monte Carlo para modelar las distribuciones a posteriori surge como opción al teorema de Bayes al obtener el cálculo de las distribuciones en la inferencia Bayesiana aproximando la integral en la ecuación (3.10) que representa la distribución a posteriori que en algunas ocasiones se vuelve intratable [29].

Método de Monte Carlo en una distribucion a posteriori Gaussiana

La distribución normal cuenta con dos parámetros para su obtención como lo es la media μ en los datos y la varianza de estos que viene dado de la desviación estándar σ:

$$p\left(\mu,\sigma^2|x\right)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]=P\left(\mu,\sigma^2\right) \tag{3.36}$$

Para el modelo normal (μ,σ^2) representan la media y la varianza respectivamente y que están disponibles de la información analizada.

Con el conocimiento del modelo de distribución a utilizar para obtener la distribución a posteriori se puede representar el cálculo de estas distribuciones en términos de probabilidades condicionas como en la ecuación (3.2) agregando los términos de los datos analizados y los parámetros del modelo de distribución seleccionado:

$$p\left(A \mid B\right) = \frac{p\left(B \mid A\right) p\left(A\right)}{p\left(B\right)} \tag{3.37}$$

En el caso de una distribución normal usando un conjunto de datos $x = \{x_1 \cdots x_n\}$ y los parámetros en la distribución $\theta = (\mu, \sigma^2)$ permite obtener la información necesaria para el cálculo de la distribución a posteriori:

$$p\left(\theta \mid x_1 \ldots x_n\right) = \frac{p\left(x_1 \ldots x_n \mid \theta\right) p\left(\theta\right)}{p\left(x_1 \ldots x_n\right)} \tag{3.38}$$

En el cálculo de distribución a posteriori como en la ecuación (3.38) se sabe que la probabilidad marginal entendiendo que $p\left(x_1 \ldots x_n\right)$ es igual a 1 por lo que la distribución a posteriori se puede aproximar $p\left(\theta \mid x\right) \propto p\left(\theta\right) \times p\left(x \mid \theta\right)$ de la siguiente forma:

$$p\left(\theta \mid x_1 \ldots x_n\right) \propto p\left(x_1 \ldots x_n \mid \theta\right) \times p\left(\theta\right) \tag{3.39}$$

Con lo que la distribución a posteriori en $p\left(\theta \mid x\right)$ se obtiene mediante una distribución de creencia a priori $p\left(\theta\right)$ actualizando mediante una función de probabilidad $p\left(x \mid \theta\right)$.

Los cálculos de la distribución a posteriori por medio del teorema de Bayes se encuentran con dificultades al realzar el cálculo de una distribución predictiva $p(\theta \mid x)$ como en la ecuación (3.10) en donde el método de Monte Carlo puede generar la aproximación de la distribución a posteriori [29].

El método de Monte Carlo puede aproximar la distribución a posteriori o predictiva por medio de la aproximación de la distribución de probabilidad a posteriori para el modelo de distribución seleccionado en los datos, como lo puede ser el modelo normal o exponencial. En el caso del modelo normal se tienen los siguientes pasos para una distribución a posteriori:

1. Definimos un número n de muestras para la variable aleatoria $x = x_1, x_2, \ldots, x_n$ los cuales tienen una distribucion normal:

$$x \mid \theta \sim N(\mu, \sigma^2) \tag{3.40}$$

donde N representa a una distribución normal o Gaussiana y θ representa los parámetros del modelo de distribución.

La distribución a posteriori del modelo normal utilizando el método de Monte Carlo se convierte en el cálculo de los parámetros del modelo normal.

2. Se propone la distribucion que tendrá la media μ como conocimiento a priori $\mu \sim N(\mu, \tau^2)$.

$$\mu \sim N(\mu_0, \tau^2) \tag{3.41}$$

La obtención de los parámetros en el modelo de la distribución μ y σ da como resultado la distribución posterior por lo que, basado en el conocimiento de la información a priori presentada en los primeros dos pasos se puede obtener la forma de los parámetros para la distribución posterior $\hat{\mu}$ y $\hat{\sigma}^2$:

$$\hat{\mu} = \frac{\tau^2}{\tau^2 + \frac{\sigma^2}{n}} + \frac{\frac{\sigma^2}{n}}{\tau^2 + \frac{\sigma^2}{n}} \tag{3.42}$$

$$\hat{\sigma}^2 = \frac{\frac{\tau^2\sigma^2}{n}}{\tau^2 + \frac{\sigma^2}{n}} \tag{3.43}$$

Lo anterior permite obtener la distribución de probabilidad a posteriori del modelo normal de la proyección de los parámetros de media μ y varianza σ^2:

$$p(\theta|x) = N\left(\frac{\tau^2}{\tau^2 + \frac{\sigma^2}{n}} + \frac{\frac{\sigma^2}{n}}{\tau^2 + \frac{\sigma^2}{n}}, \frac{\frac{\tau^2\sigma^2}{n}}{\tau^2 + \frac{\sigma^2}{n}}\right) \tag{3.44}$$

3. Finalmente, el modelo de la distribución a posteriori queda representado con los valores nuevos de media $\hat{\mu}$ y varianza $\hat{\sigma}^2$:

$$P(\theta|X) \sim N(\hat{\mu}, \hat{\sigma}^2) \tag{3.45}$$

Modelo de Monte Carlo para una distribucion a posteriori exponencial

Debido a que nos encontramos con distribuciones exponenciales en la información se opta por generar la metodología implementada para obtener las distribuciones a posteriori de este modelo siendo un contraste para la distribución de modelo normal, se utiliza el método de Monte Carlo para una distribución a posteriori con el modelo exponencial generando una distribución predictiva como en la ecuación (3.31):

$$P(X|\lambda) = \lambda e^{-\lambda X} \tag{3.46}$$

donde λ representa la media de los datos, además de que es el único parámetro para la construcción de esta forma de distribución. Con lo cual ejecutamos la metodología de Monte Carlo en los siguientes pasos para generar la distribución a posteriori:

1. Definimos una variable aleatoria $X = (x_1, x_2, ..., x_k)$ de muestras k, como en la ecuación (3.47).

$$X|\theta \sim \varepsilon(\theta) \tag{3.47}$$

En la distribución ϵ representa a la distribución de forma exponencial y θ representa a los parámetros de la distribución.

2. Se obtiene una distribución a priori usando el modelo de distribución Gamma, debido a que la distribución exponencial es un caso especial de la distribución Gamma:

$$\theta \sim \Gamma(\alpha, \beta) \tag{3.48}$$

Con lo que la distribución posterior de modelo exponencial se puede representar mediante el modelo de distribución Gamma:

$$\Gamma(\lambda|\alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\lambda\beta} \tag{3.49}$$

Por lo que la distribución posterior en el modelo exponencial puede representarse en términos de las observaciones:

$$P(\theta|X) = L(\theta)\Gamma(\lambda|\alpha,\beta)P(\theta|X) = \theta^{(\alpha+k)+1}e{-}\lambda(\beta + k\bar{X}) \tag{3.50}$$

donde $\theta^{(\alpha+k)+1}$ representa la función de actualización o verosimilitud, mientras que $e{-}\lambda(\beta + k\bar{x})$ representa la distribución a priori.

3. Se obtiene el modelo a posteriori para la distribución exponencial de la siguiente forma:

$$P(\theta|X) \sim \gamma(\theta|\alpha + 1, \beta + k\bar{X}) \tag{3.51}$$

3.3. Distribución a posteriori basada en una actualización con datos recientes.

Basado en la idea de la obtención de las distribuciones a posteriori se genera una metodología que realice el cálculo de las distribuciones a posteriori mediante el uso de datos que contienen información reciente. Este procedimiento se ejecuta con el objetivo de mostrar la efectividad en el cálculo de la distribución a posteriori por medio de eventos con información nueva mejorando la predicción de las distribuciones de probabilidad debido a que los parámetros encontrados en la informacion reciente encuentran más cerca de las distribuciones a predecir.

La idea es implementar una metodología basada en la inferencia Bayesiana que genera información nueva para el cálculo de las distribuciones a posteriori por medio de datos recientes, así como la implementación del método de Monte Carlo para el cálculo de las distribuciones a posteriori por medio de la informacion reciente. Los resultados de esta metodología se aplicó a parámetros de sismos ocurridos en Italia.

3.3.1. Distribución a posteriori basada en una actualización con datos recientes.

El uso de observaciones que provienen de información reciente puede mejorar los resultados del modelado de las distribuciones de probabilidad, por lo que se busca que la actualización de la distribución a priori se realice con los conocimientos más recientes de los parámetros en el modelo de distribución. En la figura 3.3 se puede observar la forma de implementación la metodología para calcular la distribución a posteriori usando datos recientes.

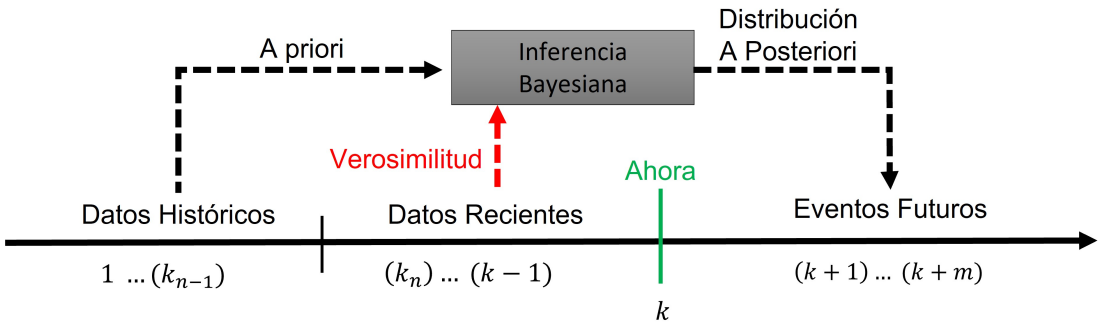


Figura 3.3: Cálculo de una distribución a posteriori con información reciente.

El calculo de la distribución a posteriori con los datos recientes estará dada por la información histórica en X_1 y la información reciente como se observa en la figura 3.3:

$$\begin{aligned} p\left(\theta \mid X_1, X_2\right) &= \tfrac{p(X_2|\theta)p(\theta|X_1)}{p(X_1,X_2)} \\ &\propto p\left(X_2 \mid \theta\right) \times p\left(\theta \mid X_1\right) \end{aligned} \tag{3.52}$$

donde $X_1 = \left[x_1 \cdots x_{k_{n-1}}\right]$ representa a la información histórica y $X_2 = \left[x_{k_n} \cdots x_{k_{-1}}\right]$ representa a la información reciente como lo podemos observar en la figura 3.3, además de que la aproximación en la distribucion a posteriori $\propto$ está dada por el conocimiento de la probabilidad marginal en $p(X_1, X_2)$.

El procedimiento de actualización por medio de datos recientes en la inferencia Bayesiana representa a un proceso en lotes [50]. El modelo es ejecutado hasta el tiempo k, después se utiliza la nueva información para actualizar la distribución utilizando los datos $X_3 = [x_{k+1} \cdots x_{k+m}]$ para obtener una de actualización.

$$\begin{aligned} p(\theta \mid X_2, X_3) &= \alpha \times p(X_3 \mid \theta) \times p(\theta \mid X_2) \\ p(\theta \mid X_2) &= p(\theta \mid X_1, X_2) \\ p(\theta \mid X_1, X_2) &= \alpha \times p(X_2 \mid \theta) \times p(\theta \mid X_1) \end{aligned} \tag{3.53}$$

En la metodología de actualización $1 > \alpha > 0$ representa el índice de entrenamiento para el calculo de distribuciones a posteriori. Con lo que al obtener la distribución posterior en los pasos de los datos en $p(\theta \mid X_1, X_2)$, se tiene una nueva información a priori $p(\theta \mid X_1)$ para ejecutar el nuevo cálculo de la distribución a posteriori por medio de los datos recientes.

Con los resultados de nuevo método de entrenamiento basado en datos recientes para la inferencia Bayesiana se pueden generar los pasos a desarrollar por medio de la metodología de Monte Carlo.

Distribución a posteriori por medio de una actualización con datos recientes para el modelo normal:

Para el calculo de la distribución a posteriori se generan los pasos para modelar la distribución predictiva en el modelo normal por medio del método de Monte Carlo en el cual la distribución se modela con un primer paso basado en los datos recientes X_1 y después obtener la distribución en un segundo paso para los datos históricos X_2.

1. Se obtiene un número n de muestras para la variable aleatoria para X_1 representando los datos históricos con un modelo normal:

$$x \mid \theta \sim N(\mu, \sigma^2) \tag{3.54}$$

donde N representa a una distribución normal o Gaussiana y θ representa los parámetros del modelo de distribución.

2. Se calcula la distribución a posteriori con una distribución de la media a priori $\mu \sim N(\mu_0, \tau^2)$ para obtener los parámetros $\hat{\mu_1} = \frac{\tau^2}{\tau^2+\frac{\sigma^2}{n}} + \frac{\frac{\sigma^2}{n}}{\tau^2+\frac{\sigma^2}{n}}$ y $\hat{\sigma}^2 = \frac{\frac{\tau^2\sigma^2}{n}}{\tau^2+\frac{\sigma^2}{n}}$ por medio del método de Monte Carlo dando como resultado la distribución para $p(\theta \mid X_1)$:

$$p(\theta \mid X_1) \sim N(\hat{\mu}, \hat{\sigma}^2) \tag{3.55}$$

3. La distribución a posteriori se calcula usando información reciente mediante la modificación de la inferencia:

$$p\left(\theta \mid X_1, X_2\right) = \frac{p\left(X_2 \mid \theta\right) p\left(\theta \mid X_1\right)}{p\left(X_1, X_2\right)} \propto p\left(X_2 \mid \theta\right) \times p\left(\theta \mid X_1\right) \tag{3.56}$$

donde $p\left(\theta \mid X_1\right)$ representa una nueva información a priori que proviene de una primera etapa de inferencia y generar los nuevos pasos del método de Monte Carlo para obtener la distribucion a posteriori $p(\theta \mid X_1, X_2)$.

Para esta actualización en la distribución a posteriori el conocimiento a priori se obtiene de los parámetros $\hat{\mu_1}$ y $\hat{\sigma_1}^2$ con la distribución del parámetro de la media $\mu_1 \sim N(\mu_{0,2}, \tau_2^2)$ con n_2 como el numero de muestras para esta etapa:

$$\hat{\mu_2} = \frac{\tau_2^2}{\tau_2^2 + \frac{\sigma_2^2}{n_2}} + \frac{\frac{\sigma_1^2}{n_2}}{\tau_2^2 + \frac{\sigma_1^2}{n_2}} \tag{3.57}$$

$$\hat{\sigma}_2^2 = \frac{\frac{\tau_2^2\sigma_1^2}{n_2}}{\tau_2^2 + \frac{\sigma_1^2}{n_2}} \tag{3.58}$$

Obteniendo la distribución de probabilidad a posteriori del modelo normal por medio del calculo de los nuevos parametros de media μ_2 y varianza σ_2^2 en el modelo.

$$(\theta|X_1, X_2) = N\left(\frac{\tau_2^2}{\tau_2^2 + \frac{\sigma_1^2}{n_2}} + \frac{\frac{\sigma_1^2}{n_2}}{\tau_2^2 + \frac{\sigma_1^2}{n_2}}, \frac{\frac{\tau_2^2\sigma_1^2}{n_2}}{\tau_2^2 + \frac{\sigma_1^2}{n_2}}\right) \tag{3.59}$$

4. Finalmente, el modelo de la distribución a posteriori queda representado con los valores nuevos de media y varianza:

$$P(\theta|X_1, X_2) \sim N(\hat{\mu}_2, \hat{\sigma}_2^2) \tag{3.60}$$

Distribución a posteriori por medio de una actualización con datos recientes para el modelo exponencial:

Se aplica el método de Monte Carlo para una distribución exponencial por medio de la obtención de una distribución predictiva como en la distribución exponencial (3.31) obteniendo una distribución en el primer paso de análisis con los datos históricos X_1 y una distribución a posteriori con la aplicación de los datos recientes X_2.

Se define a X como una variable aleatoria independiente e idénticamente distribuida en la cual se supone un modelo de su distribución de forma exponencial.

$$p(X|\lambda) = \lambda e^{-\lambda X} \tag{3.61}$$

El modelo exponencial sólo tiene la variable de la media como su único parámetro presente en los cálculos. Con base en el cálculo de la distribución a posteriori del modelo exponencial se puede generar el procedimiento para la metodología de inferencia con datos recientes.

$$p(\lambda|X) = \frac{p(X|\lambda)p(\lambda)}{p(X)}, \quad p(\lambda|X) \propto p(X|\lambda)p(\lambda) \tag{3.62}$$

Debe recordarse que en la distribución a posteriori se tiene que $p(X|\lambda)$ y $p(\lambda)$ representan las funciones de verosimilitud y la información a priori respectivamente. Con lo que se aplica el método de Monte Carlo para la búsqueda de la distribución de probabilidad añadiendo el método de entrenamiento utilizando información reciente.

El proceso de Monte Carlo para la inferencia por medio de datos recientes puede obtenerse mediante los siguientes pasos:

1. Generar dos conjuntos de datos con muestras aleatorias como $X_1 = [x_1 \cdots x_{k-n-1}]$ y $X_2 = [x_{k-n} \cdots x_{k-1}]$:

$$X_1|\theta \sim \epsilon(\theta_1) \qquad X_2|\theta \sim \epsilon(\theta_2) \tag{3.63}$$

donde ϵ representa el modelo exponencial y θ los parámetros de la distribución, como en el caso simple de la inferencia Bayesiana se utiliza la distribución Gamma como herramienta para la información a priori de θ:

$$\theta_2 \sim \gamma(\alpha, \beta) \tag{3.64}$$

2. La distribución a posterirori queda representada en términos de la distribución Gamma:

$$\gamma(\lambda|\alpha, \beta) = \frac{\beta^\alpha}{\gamma(\alpha)} \lambda^{\alpha-1} e^{-\lambda\beta} \tag{3.65}$$

3. La distribución a posteriori se calcula usando información reciente mediante la modificación de la inferencia:

$$p\left(\theta \mid X_1, X_2\right) = \frac{p\left(X_2 \mid \theta\right) p\left(\theta \mid X_1\right)}{p\left(X_1, X_2\right)} \propto p\left(X_2 \mid \theta\right) \times p\left(\theta \mid X_1\right) \tag{3.66}$$

$$p(\lambda|X_1, X_2) = L(\theta|X_2) \times \gamma(\lambda|\alpha, \beta, X_1) \tag{3.67}$$

De lo anterior se obtiene la inferencia Bayesiana utilizando datos en términos de las observaciones como:

$$p(\lambda|X_1, X_2) = \theta^{(\alpha+k)-1} e^{\left(-\lambda(\beta+k\bar{X})\right)} \tag{3.68}$$

En donde se tiene que la parte $\theta^{(\alpha+k)-1}$ representa la función de verosimilitud que proviene de X_2, mientras que la información a prior $e^{\left(-\lambda(\beta+k\bar{X})\right)}$ proviene de la información en X_1.

Con la distinción de que el proceso de actualización o la función de verosimilitud proviene de información reciente.

4. La distribución Posterior que usa información reciente en el modelo exponencial es calculada de la siguiente forma:

$$p(\lambda|X_1, X_2) = \gamma(\theta|\alpha + k, \beta + k\bar{X}) \tag{3.69}$$

El cálculo de las distribuciones a posteriori para los modelos de distribución normal o exponencial utilizando información reciente en la actualización de la información se aplicó a los parámetros sísmicos de una región de Italia, los resultados se pueden observar en la sección 5.1.1 de este trabajo.

Capítulo 4

Modelado de series de tiempo mediante redes neuronales e Inferencia Bayesiana.

En esta parte del trabajo se plantea la idea de combinar la metodología sobre inferencia Bayesiana y la información estadística de los datos analizados por medio de una red neuronal para agregar esta información en los procedimientos de la red neuronal para mejorar las tareas de modelo de distribución de probabilidad y de identificación de sistemas dinámicos, generando combinaciones entre estructuras para modelar distribuciones de probabilidad y modelar sistemas dinámicos.

Tanto la herramienta de inferencia Bayesiana como las redes neuronales pueden generar un modelo utilizando solo una parte de información de los sistemas analizados sin la necesidad de tener un modelo de dichos sistemas [56][39] [10]. La mayoría de las estructuras de las redes neuronales no utilizan la información estadística que puede estar contenida en los mismos datos analizados, por lo que la combinación de las estructuras de red neuronal para modelar distribuciones de probabilidad y para modelado de sistemas dinámicos se realiza con base en que se tome en cuenta la información estadística de los datos analizados.

Se puede aprovechar la información estadística en los datos para modelar sistemas

dinámicos por medio de una red neuronal [17], o en el caso de las distribuciones posteriores se pueden aplicar las redes neuronales para modelar distribuciones de probabilidad de los datos analizados y combinar con la inferencia Bayesiana para obtener distribuciones de probabilidad a posteriori.

En el enfoque probabilista de las redes neuronales se han presentado trabajos que realizan la inferencia Bayesiana junto con redes neuronales para el cálculo de distribuciones de probabilidad por medio del algoritmo de Mínimos Cuadrados [17], o combinaciones en la estructuras de la red neuronal para la densidad de mezcla [7], y enfoques variacionales para el modelado de distribuciones [46]. Las tareas realizadas por estas herramientas no se enfocan directamente en la identificación de sistemas. Mientras que por otro lado, encontramos estructuras específicamente enfocadas al cálculo de probabilidades como lo son las máquinas restringidas de Boltzmann (Resticted Boltzmann Machine RBM) que se enfocan en la probabilidad de un parámetro del sistema a modelar[19], [27].

La combinación de las redes neuronales y la información Bayesiana se ha implementado principalmente en tareas de distribuciones de probabilidad [22] o la aplicación de las redes enfocarse a tareas de reconocimiento de las distribuciones de probabilidad [68][43].

Otro enfoque a realizar es la combinación de redes neuronales con la informacion estadistica que se encuentra de la informacion disponible para modelar sistemas dinámicos como se presenta en la sección 4.1.2 de este trabajo.

4.1. Inferencia Bayesiana generada mediante una red neuronal.

La primera tarea por desarrollar es generar la inferencia Bayesiana con la ayuda de una red neuronal o combinaciones de estos métodos para la obtención del modelo de distribución de probabilidad en un sistema analizado. Con estas combinaciones se busca mejorar las

predicciones de las distribuciones realizadas en la inferencia Bayesiana debido a la interacción con la red neuronal.

La idea es implementar una red neuronal que modela una distribución de probabilidad implementada para generar la función de distribución $p(x \mid \theta)$ por medio de una red de mezcla de densidad o *Mixture Density Networks (MDN)* con el cual se actualiza a la distribución a priori $p(\theta)$ para aproximar una distribución a posteriori $p(\theta \mid x)$ en la inferencia Bayesiana. Primero se desarrolla la teoría para implementar una red neuronal para modelar distribuciones de probabilidad y después se plantea la idea y metodología a seguir en la aplicación de la red neuronal al cálculo de una distribución a posteriori $p(\theta \mid x) \propto p(x \mid \theta)p(\theta)$.

4.1.1. Red neuronal para modelar distribuciones de probabilidad.

En las redes neuronales Bayesianas existen dos caminos a seguir: 1) Se crea un árbol de probabilidad en lo cual se calculan la probabilidades condicionales con el cual se obtienen relaciones entre las variables de la tarea analizada cómo se observa en la figura 2.14 con el ejemplo de la caries dental, 2) Se forma una estructura de red neuronal para calcular el modelo de distribución de probabilidad que se tiene en la tarea analizada. En este trabajo nos enfocamos en las redes neuronales para generar las distribuciones de probabilidad *MDN*, en otras palabras, la opción 2.

En la arquitectura de estas redes neuronales para distribuciones de probabilidad (*MDN*) la meta es obtener el modelo de distribución de probabilidad que se tiene en los datos $p(t \mid x)$ donde x representa a las variables de entrada en la red y t representa al vector objetivo en la tarea como lo puede ser la salida del sistema analizado. En este tipo de estructuras se trabaja con los modelos de distribuciones Gaussianas:

$$p\left(t \mid x\right) = N\left(t \mid y\left(x, \omega\right), b^{-1}\right) \tag{4.1}$$

Por lo que el modelo de la red neuronal queda representado por los parámetros a modificar en la red neuronal, como lo pueden ser los pesos, el número de neuronas, etc., generando un

número L de componentes:

$$p\left(t \mid x\right) = \sum_{j=1}^{L} \pi_i\left(x\right) N\left(t \mid \mu_i\left(x\right), \sigma_i^2\left(x\right)\right) \tag{4.2}$$

Obteniendo una estructura para la red con el parámetro de mezcla en la densidad π_i, y los parámetros μ_i que representa el parámetro de la media y σ_i^2 como la varianza que provienen del modelo de distribución normal. Con lo que la tarea de la red neuronal se convierte en una predicción de los parámetros del modelo normal para lograr el modelo de la distribución de probabilidad $p(t \mid x)$. En la figura 4.1 se muestra el bosquejo del objetivo de la red neuronal para producir distribuciones de probabilidad $p(t \mid x)$ dado un conjunto de datos x.

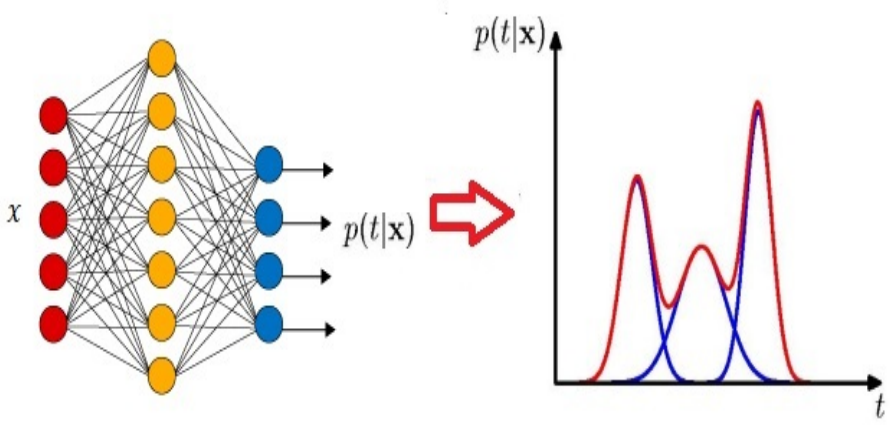


Figura 4.1: Distribución de probabilidad modelada mediante una red neuronal.

Para obtener la salida de la distribución de probabilidad por medio de la red neuronal se aplica una red de densidad de mezcla la cual tiene dos etapas; la primera es generar los datos por medio una red neuronal, y el paso dos es generar el modelo de la densidad. En la figura 4.2 se muestran las dos etapas a seguir por parte de la red neuronal Bayesiana para obtener distribuciones de probabilidad.

La estructura de la red para determinar los parámetros θ está constituida por una capa

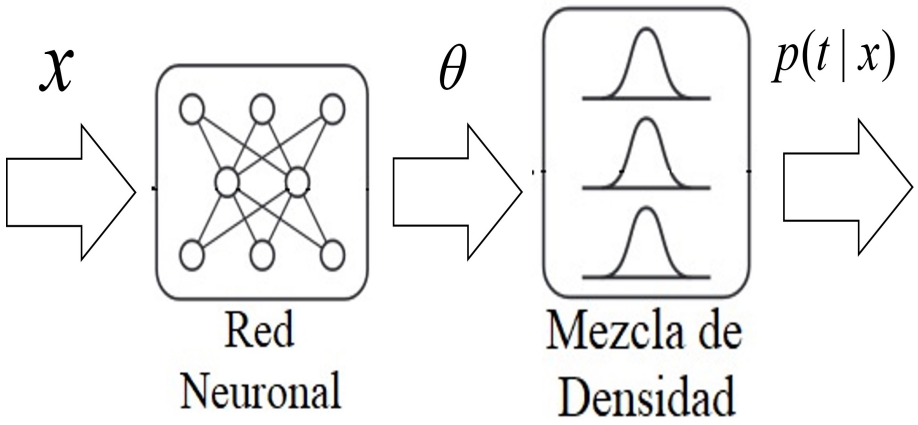


Figura 4.2: Red neuronal para obtener Distribuciones de probabilidad (*Mixture Density Network*).

oculta y una capa de salida definidas por las siguientes ecuaciones:

$$\begin{aligned} h_1 &= max\left(W_1^T x + b_1, 0\right) \\ \pi &= softmax(W_\pi^T h_1 + b_\pi) \\ \mu &= \left(W_\mu^T h + b_\mu\right) \\ \sigma &= \left(W_\sigma^T h + b_\sigma\right) \end{aligned} \tag{4.3}$$

donde la salida de la red entrega a los parámetros de la densidad π, μ y σ, con los pesos de la entrada a la capa oculta W_1, W_π representa a los pesos entre la capa oculta y el parámetro de mezcla de densidad, W_μ son los pesos entre la capa oculta y la salida del parámetro de la media μ y W_σ representa a los pesos en la capa oculta hacia la salida del parámetro σ.

La salida de la red neuronal para distribuciones de probabilidad forma una combinación de una red neuronal para los datos del modelo junto con el modelo de la distribución de probabilidad:

$$p\left(t \mid x\right) = \sum^{m} \pi_i\left(x\right) \phi\left(t \mid x\right) \tag{4.4}$$

En donde la función de ϕ representa el modelo de probabilidad aplicado a la red neuronal:

$$\phi_i\left(t \mid x\right) = \frac{1}{\left(2\pi\right)^{c/2}\sigma_i(x)^c} \exp\left\{\frac{\left\|t - \mu_i\left(x\right)\right\|^2}{2\sigma_i(x)^2}\right\} \tag{4.5}$$

En el caso de las redes neuronales para distribuciones de probabilidad se tiene el error con el que se podrá evaluar la densidad modelada que depende directamente del modelo de la distribución de probabilidad y no del error de modelado como en una red neuronal para identificación:

$$E^q = -\ln\left\{\sum_{i=1}^{m} \pi_i\left(x^q\right)\phi_i\left(t^q \mid x^q\right)\right\} \tag{4.6}$$

Se introduce el parámetro de la mezcla α dentro del modelo de la ecuación (4.5) cumple con las siguientes propiedades:

$$\sum_{i=1}^{L} \pi_i\left(x\right) = 1, \quad 0 \leq \pi_i\left(x\right) \leq 1 \tag{4.7}$$

La salida del modelo en la red neuronal para modelar una distribución pasa por el proceso *softmax* con lo cual el rango de la salida se encuentra en $[0, 1]$ y se consigue cumplir con las propiedades para la mezcla de densidad π:

$$\pi_i\left(x\right) = \frac{\exp\left(\theta_i^{\pi}\right)}{\sum_{l=1}^{L} \exp\left(\theta_i^{\pi}\right)} \tag{4.8}$$

donde θ_i^{α} representa a la salida de la red. Debido a los parámetros de la distribución Gaussiana en la red neuronal para modelar distribuciones el procedimiento de entrenamiento en la red también cumple con propiedades para el parámetro de la media μ:

$$\mu_{ij}(x) = \theta_{ij}^{\mu} \tag{4.9}$$

En donde $\mu_{ij}(x)$ representa a los componentes del núcleo (*Kernel*) en los centros $\mu_i(x)$, θ_{ij}^{μ}

representa el número de salidas $i * j$.

De la misma forma la red neuronal para modelar distribuciones por medio del modelo normal cumple las siguientes propiedades para el parámetro de varianza σ_l^2:

- $\sigma_i^2(x) \geq 0$.
- Con activaciones exponenciales se tiene que $\sigma_i(x) = \exp(\theta_i^\sigma)$.

La estructura de la red neuronal para modelar la distribución de probabilidad en términos del modelo de distribución normal es:

$$p\left(t \mid x\right) = \sum_{l=1}^{L} \pi_i\left(x\right) N\left(t \mid \mu_i\left(x\right), \sigma_i^2\left(x\right)\right) \tag{4.10}$$

El modelo en la ecuación (4.10) puede representarse en términos de los pesos de la red neuronal para modelar distribuciones:

$$E\left(\omega\right) = -\sum_{n=1}^{N} \ln\left\{\sum_{l=1}^{L} \pi_i\left(x_n, \omega\right) N\left(t \left| \mu_i\left(x_n, \omega\right), \sigma_i^2\left(x_n, \omega\right)\right.\right)\right\} \tag{4.11}$$

Para poder realizar las derivada del error en la ecuación (4.11) se presentan las dependencias con respecto de los parámetros del modelo normal, pero antes de realizar las derivada se puede representar al modelo de la red neuronal para modelar distribuciones como el calculo de la distribución a posteriori:

$$\gamma_k\left(t \left| x\right.\right) = \frac{\pi_i N_i}{\sum_{i=1}^{k} \pi_i N_i} \tag{4.12}$$

donde γ_k denota la distribución posterior con los parámetros $N(t_n \mid \mu_k(x_n), \sigma_k^2(x_n))$. Ademas los coeficientes en $\pi_i(x)$ representan probabilidades a priori dependientes de la entrada x.

Obteniendo las derivadas del error con la dependencia de cada uno de los parámetros del modelo de la distribución normal:

- Componente de mezcla de la distribución:

$$\frac{\partial E_n}{\partial \theta_k^{\pi}} = \pi_k - \gamma_k \tag{4.13}$$

- Media:

$$\frac{\partial E_n}{\partial \theta_{ki}^{\mu}} = \gamma_k \left\{ \frac{\mu_{ki} - t_i}{\sigma_k^2} \right\} \tag{4.14}$$

- Varianza:

$$\frac{\partial E_n}{\partial \theta_{ki}^{\sigma}} = -\gamma_k \left\{ \frac{\|t - \mu_k\|^2}{\sigma_k^2} - \frac{1}{\sigma_k} \right\} \tag{4.15}$$

Una vez entrenada la red neuronal para modelar distribuciones se puede predecir la función de distribución de probabilidad $p(t \mid x)$ por medio de los datos objetivo t dado el valor de las entradas x. Con esto se puede obtener los valores de los parámetros para la media μ condicional a los datos objetivo t, por medio de la esperanza $\mathbb{E}$ y la varianza $s^2(x)$ sobre la media condicional [7]. El promedio de la distribución objetivo se puede obtener mediante la esperanza $\mathbb{E}$:

$$\mathbb{E}\left[t \,|x\right] = \int tp\left(t \,|x\right) dt = \sum_{l=1}^{L} \pi_i\left(x\right) \mu_i\left(x\right) \tag{4.16}$$

Encontrando la varianza para la media condicionada:

$$\begin{aligned} s^2\left(x\right) &= \mathbb{E}\left[\left\|t - \mathbb{E}\left[t \,|x\right]\right\|^2 |x\right] \\ s^2\left(x\right) &= \sum_{l=1}^{L} \pi_i\left(x\right) \left\{ \sigma_i^2\left(x\right) + \left\| \mu_i\left(x\right) - \sum_{l=1}^{L} \pi_i\left(x\right) \mu_i\left(x\right) \right\|^2 \right\} \end{aligned} \tag{4.17}$$

4.1.2. Entrenamiento de una red neuronal para obtener distribuciones de probabilidad.

En este proceso se busca minimizar la función de la suma de los cuadrados del error:

$$E^{LS}(W) = \frac{1}{2}\sum_{q=1}^{n}\sum_{k=1}^{c}[f_k(x^q;W) - t_k^q]^2 \tag{4.18}$$

En este tipo de entrenamiento se define el error E^{LS}. El error del modelo se representa en términos de las distribuciones de probabilidad mediante una integral que contiene a la probabilidad conjunta de los datos x y el vector objetivo t como $p(x,t) = p(t \mid x)p(x)$:

$$\begin{aligned} E^{LS} &= \lim_{n\to\infty} \tfrac{1}{2n}\textstyle\sum_{q=1}^{n}\sum_{k=1}^{c}[f_k(x^q;W) - t_k^q]^2 \\ E^{LS} &= \tfrac{1}{2}\textstyle\sum_{k=1}^{c}\iint [f_k(x^q;W) - t_k^q]^2 p\,(t,x)\,dtdx \end{aligned} \tag{4.19}$$

donde t_k es el vector objetivo de la forma $t = (t_1, ..., t_c)$, $f_k(x^q;W)$ es el conjunto de salida en la red neuronal y x es el conjunto de entradas en la red neuronal de la forma $x = (x_1, ..., x_n)$. En donde se busca minimizar el error por medio de la derivada de la función del error con respecto a la red neuronal $f_k(x^q;W)$:

$$\frac{\delta E^{LS}}{\delta f_k(x^q;W)} = 0 \tag{4.20}$$

La expresión para minimizar la ecuación (4.20) por medio del ajuste de los pesos W en la red f_k se representa de la siguiente manera:

$$f_k\,(x, W^*) = \langle t_k \,|x\rangle \tag{4.21}$$

donde W^* representa a los pesos obtenidos por medio del proceso de minimización. Se define a la media condicional para una cantidad Q:

$$\langle Q\,|x\rangle = \int Q\,(t)\,p\,(t\,|x\,)\,dt \tag{4.22}$$

El resultado de entrenar la red neuronal f_k por medio de Mínimos Cuadrados aproxima a los parámetros estadísticos de los datos objetivo, la media que se tiene con respecto del vector de entrada x y la varianza media de los datos con respecto del promedio condicionado. Si conocemos los paramentos estadísticos podemos representar la distribución condicional de

los datos objetivo por medio del modelo Gaussiano con centro obtenido mediante f_k con un parámetro de varianza global. El uso de mínimos cuadrados no asume que la distribución en los datos objetivo sea normal pero tampoco puede distinguir entre una distribución normal y otro tipo de distribución con las mismos parámetros de media condicional y varianza global [7].

Si se define a la distribución condicional de los datos objetivo como Gaussiana obtenemos el formalismo del método de Mínimos Cuadrados al emplear la máxima verosimilitud con σ como la varianza global:

$$p\left(t\left|x\right.\right) = \frac{1}{\left(2\pi\right)^{1/2}\sigma}\exp\left\{-\frac{\left|F_k\left(x\right)-t_k\right|^2}{2\sigma^2}\right\} \tag{4.23}$$

F_k representa la media de los datos objetivo de t_k y es una función general de x. La distribución condicional del vector objetivo completo es:

$$p\left(t\left|x\right.\right) = \prod_{k=1}^{c} p\left(t_k\left|x\right.\right) = \frac{1}{\left(2\pi\right)^{c/2}\sigma^c}\exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{c}\left|F_k\left(x\right)-t_k\right|^2\right\} \tag{4.24}$$

En el entrenamiento de la red se tiene el objetivo de determinar la cantidad F_k por medio de $f_k(x, W)$ al maximizar la verosimilitud del conjunto de datos (x^q, t^q), Si asumimos que el conjunto de datos entrenados se dibujan independientemente con el modelo dado en la ecuación (4.24) entonces la verosimilitud del conjunto de datos se obtiene por el producto de la verosimilitud de cada dato [7]:

$$\mathcal{L} = \prod_{q=1}^{n} p\left(t^q, x^q\right) = \prod_{q=1}^{n} p\left(t^q\left|x^q\right.\right) p\left(x^q\right) \tag{4.25}$$

Se obtienen los valores adecuados para W maximizando a $\mathcal{L}$. En la práctica se prefiere minimizar el logaritmo negativo de $\mathcal{L}$ definida como la función del error:

$$\mathcal{E} = -\ln\mathcal{L} \tag{4.26}$$

donde $\mathcal{E}$ es la función del error para la actualización de los parámetros de la red neuronal. Y esto se debe a que maximizar $\mathcal{L}$ en la ecuación (4.26) sería equivalente a minimizar la función del error $\mathcal{E}$, con el conocimiento de que la función logaritmo negativo es monótona [7].

Al obtener los valores de los parámetros W^* se puede obtener el valor óptimo de σ que minimiza a $\mathcal{E}$ con respecto de σ:

$$\sigma^2 = \frac{1}{nc} \sum_{q=1}^{n} \sum_{k=1}^{c} \left[f_k \left(x^q; \omega^* \right) - t_k^q \right]^2 \tag{4.27}$$

Los parámetros W^* se obtienen a través del método de Mínimos Cuadrados el cual se analiza en la sección 4.3.1 de este trabajo.

4.2. Combinación de la inferencia Bayesiana y la red neuronal para obtener distribuciones de probabilidad.

La combinación de la red neuronal para modelar una distribución de probabilidad con la inferencia Bayesiana surge de la idea de obtener distribuciones a posteriori, además de la idea de obtener una mejor distribución a posteriori por medio de información reciente en los datos.

Retomando el modelo de inferencia Bayesiana por medio de las probabilidades condicionales de modelos de distribución como en la ecuación (4.28) :

$$\begin{aligned} p\left(\theta \mid x\right) = p\left(x \mid \theta\right) p\left(\theta\right) = \tfrac{p(x|\theta)p(\theta)}{\sum p(x)} \\ \propto p\left(x \mid \theta\right) p\left(\theta\right) \end{aligned} \tag{4.28}$$

donde $p\left(\theta \mid x\right)$ representa a la distribución a posteriori, $p\left(\theta\right)$ es la distribución a priori y $p\left(x \mid \theta\right)$ la distribución de probabilidad que actualiza la distribución a priori. En esta última parte del método de inferencia Bayesiana se implanta la estimación de la distribución de

actualización por medio de una red neuronal para modelado de distribuciones de probabilidad *MDN*. Por lo que el modelo obtenido para $p(t \mid x)$ en la red neuronal representara a la distribución de probabilidad $p(x \mid \theta)$ en la inferencia Bayesiana:

$$p(t \mid x) = \sum_{j=1}^{L} \pi_j(x) N\left(t \mid \mu_j(x), \sigma_j^2(x)\right) \tag{4.29}$$

Con lo que la red neuronal *MDN* encuentra la función de distribución $p(t \mid x)$ maximizando la distribución del conjunto de parámetros (t, x) y esta distribución se aplica como $p(x \mid \theta)$ en el cálculo de la distribución a posteriori:

$$\mathcal{E} = -\ln \mathcal{L} \tag{4.30}$$

En la tabla 4.1 se aplica la metodología para obtener el modelo de la distribución a posteriori de un sistema con datos disponibles y aplicando una red neuronal para modelar distribuciones de probabilidad que obtendrá $p(t \mid y = x)$ con la cual se actualiza la distribución a priori $p(\theta)$ la cual se puede generar con el conocimiento de los datos se puede observar.

Tabla 4.1: Modelo estadístico usando redes neuronales

1. Generar la información a priori $p(\theta)$.
2. Con y calcular $p(\hat{y})$ con la MDN usando datos recientes.
3. Usar $p(\hat{y})$ como $p(x\|\theta)$ para actualizar $p(\theta)$.
4. Obtener la distribucion a posteriori $p(\theta\|X)$.

Para analizar los resultados del procedimiento de inferencia Bayesiana por medio de una red neuronal se aplica esta metodología a distribuciones de parámetros sísmicos en México en la sección 5.2 de este trabajo seleccionando un grupo de datos $X_1 = x_1, \ldots, x_{kn-1}$ de donde se calcula la distribución a prior $p(\theta)$ además de agregar un grupo de datos $X_2 = x_{kn}, \ldots, x_{k-1}$ para obtener la distribución $p(x \mid \theta)$ por medio de la red neuronal para distribuciones de probabilidad. Con este resultado se calcula la diferencia entre las distribuciones por medio de la divergencia *Kullbac-Leibler* $KL = p(y_{real}) log[p(y_{real}/p(\theta|x))]$

donde la distribución $p(y_{real})$ representa a la distribución en los datos $X_3 = x_{kn+1}, \ldots, x_{km}$ como se puede observar en la figura 3.3.

En la figura 4.3 se puede observar el modelo implementado de la combinación del método de inferencia Bayesiana con el uso de una red neuronal para modelar distribuciones de probabilidad. Con esta estructura se busca modelar la distribución del modelo que se tienen en la dinámica de salida del sistema.

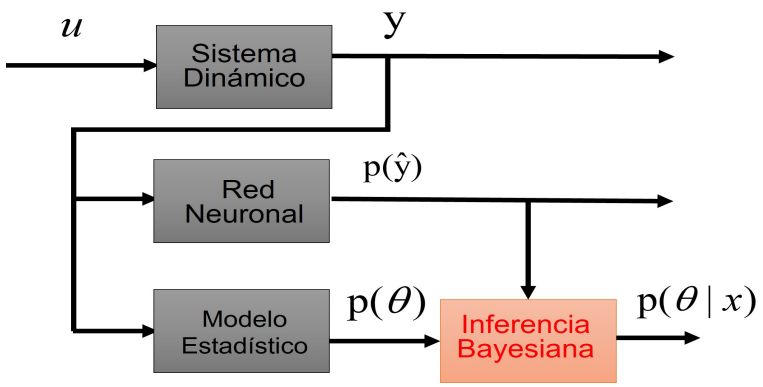


Figura 4.3: Inferencia Bayesiana modelada con red neuronal.

4.3. Modelado de un sistema dinámico por medio de redes neuronales y la información estadística de los datos disponibles.

Otro de los enfoques de este trabajo consiste en usar la información estadística x y y disponible en el sistema dinámico para emplear esta información a una red neuronal. Al agregar la información estadística a la red neuronal se emplean distribuciones de probabilidad condicionadas $p(y \mid x)$ con lo que la estructura de la red neuronal se entrena en lotes, en el primer lote se identifica el sistema dinámico con los datos disponibles de entrada y salida y el segundo lote entrena a la red con un procesos de poda en la estructura y analizando

la información estadística. La estructura de la red se puede observar en la figura 4.4. En el primer lote se entrena a la red con el método de propagación hacia atrás del error (BP), el método de mínimos cuadrados o el método de aprendizaje extremo (Extreme Learning Machines ELM) como se desarrolla en las primeras secciones 4.3.1 de este trabajo, después se desarrolla la metodología para el segundo lote de entrenamiento empleando la información estadística como se plantea en la sección 4.3.2.

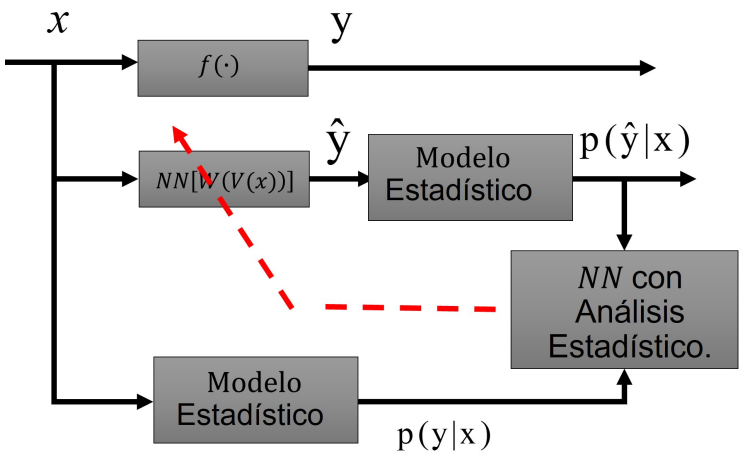


Figura 4.4: Ajuste fino en la red neuronal

4.3.1. Modelado de un sistema dinámico por medio de redes neuronales.

En las redes neuronales los sistemas se representan como modelos no lineales autorregresivo con media móvil de entradas exógenas (NARMAX) como se representa en las siguientes ecuaciones:

$$\begin{aligned} y(k) &= F\left[x\left(k\right), d\left(k\right)\right] \\ x\left(k\right) &= \left[y\left(k-1\right), \cdots y\left(k-n_y\right), u\left(k\right), \cdots u\left(k-n_u\right)\right]^T \\ d\left(k\right) &= \left[\xi\left(k\right), \cdots \xi\left(k-n_2\right)\right]^T \end{aligned} \tag{4.31}$$

Para este tipo de sistemas en la ecuación (4.31) se tienen los órdenes para la planta en las ecuaciones n_y y n_u los cuales corresponden a la entrada $u \in R^n$ y salida $y \in R^m$ del sistema a modelar para generar el vector de regresión $x \in R^{nu+ny}$ donde las dimensiones varían de sistema en sistema, aunque este conocimiento no es necesario en la red neuronal. Debido a que se desconoce con exactitud el modelo dinámico para el sistema, pero si se tienen disponibles la entrada y la salida del sistema se puede diseñar un sistema basado en redes neuronales para identificar el modelo anterior:

$$\hat{y}\,(k) = NN\,[x\,(k)\,, \omega] \tag{4.32}$$

Esta representación es un modelo de caja negra (se conocen pocas características del modelo) que se ajusta por medio de los parámetros llamados pesos ω. El termino principal para la aplicación de la red neuronal y su ajuste con respecto del modelo obtenido es el error de la salida como:

$$e\,(k) = y(k) - \hat{y}\,(k) \tag{4.33}$$

Primer lote de entrenamiento de la red neuronal.

En el primer lote de entrenamiento se ejecutan los métodos de propagación del error hacia atrás (Back Propagation BP), el método de Mínimos Cuadrados o Aprendizaje Extremo en la red neuronal.

Entrenamiento por el método de propagación del error hacia atrás.

La red neuronal necesita de una función de costo que ajuste el parámetro de los pesos ω:

$$L_1(e) = \frac{1}{2}\,[y(k) - \hat{y}\,(k)]^2 \tag{4.34}$$

Se calculan los pesos con el fin de minimizar la función de costo de la red neuronal usando una tasa de aprendizaje η en cada uno de los pasos del calculo de los pesos:

$$W_i\left(k+1\right)=W_i\left(k\right)-\eta\frac{\partial L_1(e)}{\partial W_i} \tag{4.35}$$

donde $i=1,...,m$ representa el número de pesos en la red neuronal y $L_1(e)=\frac{1}{2}e^2$ es el índice de desempeño a minimizar, En el caso de tener una capa oculta V con el número de pesos en esta capa como $j=1,...,m$ se obtiene el ajuste de sus pesos como:

$$V_{ij}\left(k+1\right)=V_{ij}\left(k\right)-\eta\frac{\partial L_1(e)}{\partial V_{ij}} \tag{4.36}$$

Entrenamiento de la red neuronal mediante el método de Mínimos Cuadrados.

Retomando el modelo de la red neuronal (4.32) tenemos su modelo representativo (ejemplo de 2 capas) para la aplicación de su entrenamiento:

$$\hat{y}=NN\left[x\left(k\right)\right]=W\phi\left[Vx\left(k\right)\right] \tag{4.37}$$

donde $x\left(k\right)\in R^{(n_u+n_y)}$ está definido en el sistema de la ecuación (4.31), $V\left(x\right)\in R^{m\times(n_u+n_y)}$, ϕ es una función de activación, el vector $W\in R^m$, $\hat{y}\in R$ con m como el número de nodos ocultos. El primer paso del entrenamiento de la red es escoger los pesos aleatoriamente en la capa oculta, obteniéndose un modelo para la salida del sistema:

$$\hat{y}\left(k\right)=W\left(k\right)\Phi\left(k\right) \tag{4.38}$$

De la ecuación (4.38) se cumple que $\Phi\left(k\right)=\left[\phi_1,\cdots,\phi_m\right]^T$. Por lo que asignando un numero q de datos para la entrada y la salida se tiene que:

$$\Phi=\left[\Phi_1\cdots\Phi_m\right],\quad \Phi_1=\left[\phi_1\left(1\right)\cdots\phi_1\left(q\right)\right],\quad Y=\left[y\left(1\right)\cdots y\left(q\right)\right] \tag{4.39}$$

Con lo que el modelo de la salida del entrenamiento queda representado en la ecuación (4.40) :

$$Y=W\Phi+E \tag{4.40}$$

donde $E = [e(1) \cdots e(q)]$, $e(k) = y(k) - \hat{y}(k)$ es el error de modelado por la red. Por lo que la función a minimizar tiene la forma $\frac{1}{2}\sum [y(k) - \hat{y}(k)]^2$ y se obtiene por medio de la inversa generalizada de Moore-Penrose la cual cumple con las siguientes definiciones:

Definición 4.1 *[30] La matriz $\Phi^+ \in \Re^{q\times m}$ es la inversa generalizada de Moore-Penrose para $\Phi \in \Re^{m\times q}$ si se cumple lo siguiente.*

$$\Phi\Phi^+\Phi = \Phi, \quad \Phi^+\Phi\Phi^+ = \Phi^+, \quad \left(\Phi^+\Phi\right)^{\mathrm{T}} = \Phi^+\Phi, \quad \left(\Phi\Phi^+\right)^{\mathrm{T}} = \Phi\Phi^+ \tag{4.41}$$

En un caso particular, cuando Φ tiene rango completo por columnas se tiene que:

$$\Phi^+ = \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^T \tag{4.42}$$

Mientras que cuando Φ sea de rango completo por filas se tiene que:

$$\Phi^+ = \Phi^T\left(\Phi\Phi^T\right)^{-1} \tag{4.43}$$

Definición 4.2 *[30] $x_0 \in \Re^q$ se dice que es una solución por Mínimos Cuadrados de la norma mínima del sistema lineal $\hat{y} = NN[x]$ si cumple con lo siguiente:*

$$\|NNx_0 - y\| = \min_x \|NNx - y\| \tag{4.44}$$

donde$\|\cdot\|$ es la norma en el espacio Euclidiano. $\hat{x}$ es solución de $\hat{y} = \Phi x$ es una solución por Mínimos Cuadrados en sentido de la norma mínima si y solo si $\hat{x} = \Phi^+\hat{y}$.

La inversa generalizada de Moore-Penrose puede minimizar la función de costo en la red neuronal, tal que:

$$W^* = Y\Phi^{\mathrm{T}}\left(\Phi\Phi^T\right)^{-1} = Y\Phi^+ \tag{4.45}$$

Por lo que W^* puede minimizar la función $\frac{1}{2}\sum [y(k) - \hat{y}(k)]^2$ en el modelo de la red neuronal (4.37).

La existencia y unicidad de la matriz Φ^+ se describe en [76].

Máquina de aprendizaje extremo (ELM) para una red neuronal.

Para el primer paso de entrenamiento en la red neuronal se utiliza un entrenamiento diferente a la propagación del error hacia atrás de la red neuronal, y con el fin de obtener una comparación de resultados, se utiliza el Aprendizaje Extremo (*Extreme learning machine ELM*) por su facilidad con la generalización en la identificación así como la rapidez en su aplicación en el entrenamiento de redes neuronales [32].

El entrenamiento simple de una estructura de red neuronal con una capa es una función para esta red representada como:

$$o_i(x) = F(\alpha_i, \beta_i, x) \tag{4.46}$$

donde α_i y β_i son parámetros de los nodos ocultos. Obteniendo la salida del entrenamiento aplicando Aprendizaje Extremo corresponde a la ecuación (4.47).

$$F_L = \sum_{i}^{L} \Phi_i O_i(x) \tag{4.47}$$

En el Aprendizaje Extremo Φ_i es el peso en cada nodo oculto. Y O representa la salida en la capa oculta de la red $O = (o_1, o_2, ..., o_L)$. Así que si se tienen j muestras de entrenamiento, se puede obtener el Aprendizaje Extremo como una matriz para la capa oculta O:

$$O = \begin{bmatrix} o\left(x_1\right) \\ \vdots \\ o\left(x_j\right) \end{bmatrix} = \begin{bmatrix} F\left(\alpha_1, \beta_1, x_1\right) & \cdots & F\left(\alpha_L, \beta_L, x_1\right) \\ \vdots & \cdots & \vdots \\ F\left(\alpha_1, \beta_1, x_j\right) & \cdots & F\left(\alpha_L, \beta_L, x_j\right) \end{bmatrix} \tag{4.48}$$

Se tiene a T como las observaciones objetivo para el entrenamiento:

$$T = \begin{bmatrix} t_1 \\ \vdots \\ t_j \end{bmatrix} \tag{4.49}$$

El objetivo del aprendizaje extremo es minimizar el error de entrenamiento y la norma

de los pesos en la salida:

$$\Phi = \min_{\Phi} \left(\frac{1}{2} \|\Phi\|^2 + \frac{C}{2} \|O\Phi - T\|^2 \right) \tag{4.50}$$

donde C representa un parámetro de regularización. En el caso de tener solo una capa oculta en la red de la forma $\hat{y} = W\psi(Vx)$ con ψ la función de activación y el grupo de pares observados (x, y) donde $x \in R^n$ y $y \in R^m$ el aprendizaje extremo se puede aplicar de la siguiente forma:

1. Iniciar los pesos en la capa oculta V.

2.Calcular los pesos W con un conjunto de datos observados de y por medio de la pseudoinversa de Moore-Penrose $W = \psi(Vx)^+ y$.

4.3.2. Segundo lote de entrenamiento de la red neuronal.

Para el segundo lote de entrenamiento se ejecuta el modelado del sistema por medio de la información estadística disponible en los datos de entrada y salida, se hace uso de la herramienta de ajuste fino de los pesos del método de poda en el cual se genera un recorte de la estructura de la red neuronal lo que nos permite ejecutar un entrenamiento por épocas de la red neuronal para el modelado de un sistema dinámico [9]. La red neuronal pasa por un primer lote de entrenamiento para modelar al sistema dinámico y en el segundo lote de entrenamiento reajustan los pesos de la red por medio el modelado de $p(y \mid x)$.

$$\hat{y} = NN_0 \left[x\left(k\right) \right] = W\phi \left[Vx\left(k\right) \right] \tag{4.51}$$

El objetivo principal de la poda es generar un recorte en el número de los pesos en la red neuronal W_m. Para esta teoría se tienen los siguientes pasos para modificar la estructura en la red:

- Paso 1: Ejecutar el análisis de la red neuronal con W_m.

- Paso 2: Definir el número de elementos en los pesos a podarse. Lo cual indica que el valor del peso se redefine como cero. se define a d como el número de pesos a eliminar como $W_d = W_1 = W_2 = W_3 = 0$.
- Paso 3: Comparar el rendimiento de la red utilizando W_d contra el rendimiento empleado en W_m.
- Paso 4: Paro del proceso. Detener el proceso cuando el rendimiento de la red neuronal que emplea W_d sea deficiente en comparación con el rendimiento obtenido con W_m.

En la figura 4.5 se tiene una representación de la metodología de poda de las redes neuronales, observando cómo se modifica un modelo original planteado para una red neuronal y la posible reestructuración de la red para que realice su funcionamiento con menos elementos en la red.

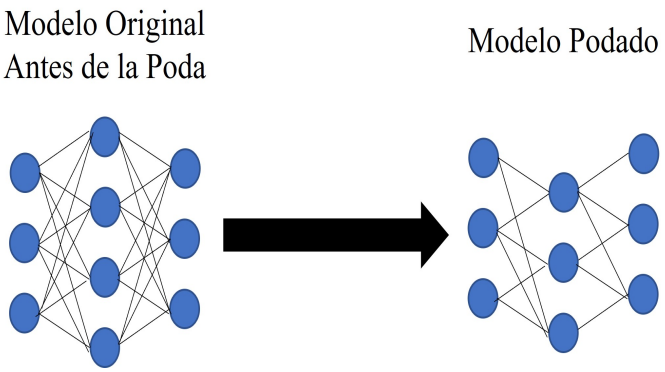


Figura 4.5: Ajuste fino en la red neuronal

Otro enfoque para la eliminación de pesos es agregar en la función de costo a minimizar un elemento que contenga el parámetro que indica el número de elementos que se quieren minimizar:

$$J = \sum_{k=1}^{m} y_k - \hat{y}_k + \varrho \sum_{c=1}^{m} \frac{W_d^2/W_m^2}{1 + W_d^2/W_m^2} \tag{4.52}$$

El segundo término $\varrho \sum_{c=1}^{m} \frac{W_d^2/W_m^2}{1+W_d^2/W_m^2}$ de la función de costo determina la complejidad del sistema basado en la suma de todos los componentes de la red neuronal y con el cual se puede medir su tamaño. El valor ϱ representa la importancia relativa del término en la función de costo. Esta metodología es aplicada directamente en la función de costo para la propagación del error hacia atrás calculando directamente los valores en la parte compleja que no afecten el rendimiento de la red neuronal.

Debido a estas estructuras nos encontramos con diferentes métodos para la poda en la red neuronal y se genera una pequeña clasificación en relación con sus diferencias entre estos.

- Estructura: En este método se podan parámetros individualmente seleccionados en la red neuronal y es conocido como poda no estructurada. Mientras que la poda estructurada elimina grupos de parámetros como un conjunto de neuronas de una capa en la red neuronal.
- Puntuación: Son métodos de poda que comparan los valores de los pesos localmente, podando una fracción de los parámetros con los valores más bajos dentro de cada capa de la red neuronal.
- Programación: Los métodos de poda difieren en la cantidad de la red a podar en cada paso. Algunos podan una fracción fija de la red de forma iterativa a lo largo de varios pasos mientras que otros métodos podan todos pesos a la vez en un solo paso.
- Ajuste fino: En este método se sigue entrenando la red utilizando los pesos entrenados antes de la poda.

En este trabajo se selecciona la metodología de ajuste fino en la poda de la red neuronal para ejecutar los dos lotes de entrenamiento para una red neuronal de dos capas representando el modelo de la red por el método de poda en la siguiente ecuación. Cuando una red se representa mediante los datos de entrada x y los parámetros de la red M como una función $NN_0 = f\,[x, M]$, a partir de este se produce una estructura de la red con ajuste fino de los pesos NN_i:

$$NN_i = f\left[x, Pr \odot M^i\right] \tag{4.53}$$

donde $\Pr \in \{0,1\}^{|M^i|}$ con M^i representa a los parámetros en la red neuronal a la cual se le aplicó la poda y puede ser diferente de M, $\odot$ es la representación del producto elemento a elemento, donde $i = 1, ..., \bar{\xi}$ que representa cada una de las época del segundo lote de entrenamiento. NN_i representa la estructura de la red neuronal después de haber realizado la poda en ella.

Para obtener NN_i se realiza el recorte de la estructura de la red por medio de la valoración de los pesos con menor valor (los pesos que están más cercanos a cero). Debido a que la red neuronal ha sido procesada con el primer lote de entrenamiento donde se obtuvo NN_0, para esta estructura calcula la probabilidad condicional $p_i(\hat{y} \mid x)$ por medio del modelo de distribución normal para variables aleatorias independientes cumpliendo que $p(x, \hat{y}) = p(x)p(\hat{y} \mid x)$:

$$p_i\left(\hat{y} \mid x\right) = \frac{p\left(x, \hat{y}\right)}{p\left(x\right)} \tag{4.54}$$

La distribución de probabilidad $p(\hat{y} \mid x)$ cumple con las propiedades de la probabilidad siguientes:

$$\begin{aligned} &\sum_x \sum_{\hat{y}} p\left(x, \hat{y}\right) = 1 \\ &E\left[x|\hat{y}\right] = \sum_x x p_i\left(\hat{y}|x\right) \end{aligned} \tag{4.55}$$

donde las distribuciones $p(y \mid x)$ se pueden calcular de la misma forma que $p_i(\hat{y} \mid x)$ utilizando la información real en los datos de entrada y salida de sistema dinámico.Una vez obtenidas las distribuciones de probabilidad $p(y \mid x)$ y $p(\hat{y} \mid x)$ se calcula la divergencia Kullback-Leibler

entre las distribuciones $p(y \mid x)$ y $p(\hat{y} \mid x)$:

$$KL_i\left(p(y\,|x)\|p(\hat{y}\,|x)\,\right) = \sum_{j}^{n} p_j(y\,|x)\log\left(\frac{p_j(y\,|x)}{p_j(\hat{y}\,|x)}\right) \tag{4.56}$$

donde n es el total de puntos en las distribuciones. La ejecución del recorte de la red neuronal y el cálculo de $KL_i(p(\hat{y} \mid x) \parallel p(y \mid x)))$ se realiza en cada época de entrenamiento y se almacenan. Además, para cada nueva estructura NN_i es necesario actualizar los pesos mediante el algoritmo de la propagación del error hacia atrás minimizando la misma de costo dada por:

$$L(e) = \frac{1}{2}\left[y(k) - \hat{y}\,(k)\right]^2 \tag{4.57}$$

La actualización de los pesos W y V que minimizan la ecuación 4.57 se obtienen como se tiene en las ecuaciones (4.35) y (4.36). Estos pesos actualizados forman al conjunto M^i, sobre los cuales se realiza la poda Pr.

Una vez que se terminan las épocas del segundo lote de entrenamiento se selecciona la mejor estructura de la red neuronal NN_O basándose en:

$$O = \arg\min_i KL_i\left(p(y \mid x) \parallel p(\hat{y} \mid x)\right). \tag{4.58}$$

En la tabla 4.2 se presenta el procedimiento a seguir para la identificación de sistemas dinámicos ejecutando los dos lotes de entrenamiento aplicado a estructuras de redes neuronales con dos capas $NN[W(Vx)]$ con $W \in R^m$ y $V \in R^{v\times m}$. En el primer lote de entrenamiento se utilizan los métodos de propagación hacia atrás del error BP o el Aprendizaje Extremo ELM, mientras que en el segundo lote de entrenamiento se realiza el recorte de la estructura, se analiza la información estadística y se entrena la red por el método BP. Esta metodología se aplica a la identificación dos sistemas dinámicos en la sección 5.3 de este trabajo.

Tabla 4.2: Ajuste fino en la red neuronal.

1. Iniciar los pesos W, V para $NN_0\left[W(Vx\left(k\right))\right]$.
2. Entrenar la red $NN_0\left[W(Vx\left(k\right))\right]$ con BP o ELM.
3. Guardar los pesos W, V.
4. Calcular $p_0(\hat{y} \mid x)$.
5. Calcular $KL_0\left(p(\hat{y} \mid x) \parallel (y \mid x)\right)$
for i=1 hasta $\bar{\xi}$
6. Podar los pesos W, V en la red $NN_{i-1}\left[W(Vx\left(k\right))\right]$.
7. Entrenar la red con BP obteniendo $NN_i\left[W(Vx\left(k\right))\right]$.
8. Calcular $p_i(\hat{y} \mid x)$.
9. Calcular $KL_i\left(p(\hat{y} \mid x) \parallel (y \mid x)\right)$.
10. Guardar NN_i, KL_i.
End for
11. Elegir la mejor estructura NN_O con base a (4.58).

Capítulo 5

Aplicaciones y Simulaciones.

En este capítulo se desarrolla la aplicación de las metodologías presentadas en los capítulos 3 y 4 de este trabajo. La primera parte del capítulo presenta la aplicación de los pronósticos en las distribuciones de probabilidad de los parámetros sísmicos por medio la inferencia Bayesiana, así como la inferencia Bayesiana actualizada con de información de datos recientes. La aplicación del modelo de la distribución de probabilidad de los parámetros sísmicos también se lleva a cabo mediante la combinación de la inferencia Bayesiana y una red neuronal para calcular distribuciones a posteriori. Por último, se modelan dos sistemas dinámicos mediante la combinación de redes neuronales y la información estadística presentadas en la sección 4.3.2 de este trabajo.

5.1. Cálculo de distribuciones a Posteriori de parámetros sísmicos en Italia.

Se analizan 4 catálogos de información sísmica de la región de central de Italia del año 1995 al año 2018, en donde uno de los catálogos representa la información completa de cada evento sísmico, mientras que los otros tres catálogos han sido procesados con técnicas de rechazo de réplicas de los eventos sísmicos en los cuales se tendrá menos datos

de informacion sísmica quedando solo los eventos sísmicos iniciales (Estas técnicas no se realizan en este trabajo pero se puede revisar el trabajo de Van shipout [78]). En este trabajo nos enfocamos en los parámetros de magnitud, distancia entre los eventos y el tiempo entre los eventos para desarrollar la predicción de las distribuciones. En la figura 5.1 se puede observar la región de interés que está comprendida entre las coordenadas 12.3°E a 13.6°E longitudinal y las coordenadas 41.6°N hasta 44°N latitudinales, donde se presentan eventos con una magnitud mayor a 5.5 en la escala de Richter [72].

Figura 5.1: Área de investigación sísmica. El recuadro marca las coordenadas de interés.

Calculo de distribuciones a posteriori para parámetros sísmicos de una región de Italia.

Se seleccionaron las fechas entre el primero de enero del 2014 a 31 de diciembre del 2016 para obtener la información con la que se construye el conocimiento a priori de las distribuciones contando con 34772 datos para calcular la distribucion a posteriori de las distribuciones de magnitud, distancia entre eventos y tiempo entre eventos, con lo cual nos

acercamos a una predicción sobre las distribuciones de estos parámetros en el año 2017 donde se tiene un total de 13784 datos de la magnitud distancia y tiempo de los eventos sísmicos. Toda esta información se encuentra disponible en `cnt.rm.ingv.it/en/iside`.

Para realizar la evaluación del modelo obtenido mediante la metodología de Monte Carlo en las distribuciones a posteriori se obtuvo la divergencia entre las distribuciones por medio de la divergencia de Kullback-Leibler, manejando el logaritmo natural en todos los cálculos de la divergencia de distribuciones obteniendo los resultados en nats:

$$KL\left[p(y_i)\,\|p(\theta\,|y_i)\,\right] = \sum_i p(y_i)\log\frac{p(y_i)}{p(\theta\,|y_i)} \tag{5.1}$$

donde la $p(\theta|y)$ representa la distribución obtenida mediante la inferencia Bayesiana y $p(y_i)$ representa la distribución real en los datos para el año 2017.

Se analizó el modelo de distribución que tiene cada parámetro sísmico en cada uno de los catálogos de información previo al cálculo de las distribuciones a posteriori seleccionando el modelo normal o exponencial y finalmente se calcula la divergencia de Kullback-Leibler entre las distribuciones. Para el catálogo sin rechazo de replicas se seleccionó el modelo de distribución exponencial para calcular las distribuciones a posteriori.

En la figura 5.2 se tiene el comportamiento del error de cada uno de los parámetros analizados contra el número de muestras k aplicadas en modelo de las distribuciones de probabilidad a posteriori por medio del método de Monte Carlo encontrando la divergencia de Kullback-Leibler con base e resultando la divergencia en nats.

El cálculo de las distribuciones a posteriori basada en información histórica disponible como información previa, en el caso de los parámetros sísmicos de Italia son los datos entre los años 2014 y 2016, nos permite comparar el resultado con la distribución que se tiene en el año 2017 como análisis de la inferencia Bayesiana. El modelo de las distribuciones a

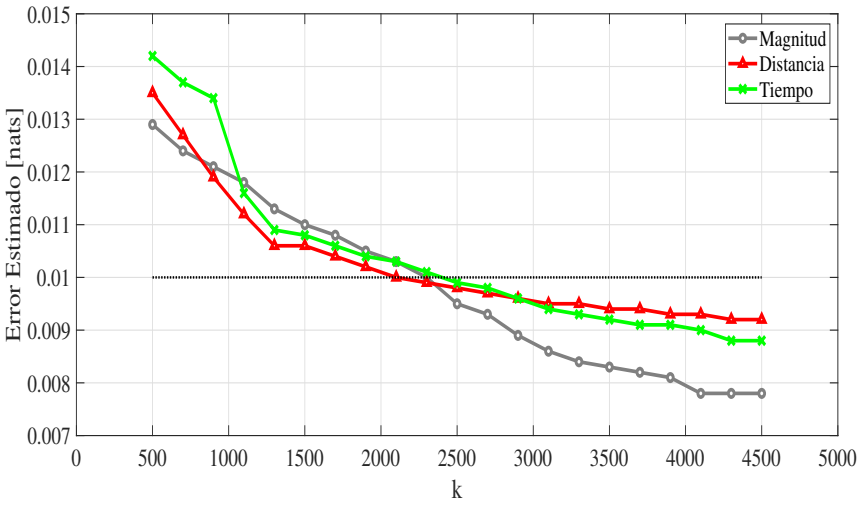


Figura 5.2: Resultados del error del catalogo principal

posteriori se calcula con el método de Monte Carlo en el cual se depende de un numero de muestras k por lo que primero se analizó la divergencia entre distribuciones con respecto al número de muestras para obtener una divergencia menor o igual a el 10 por ciento, esto se representa con una línea en las figuras del análisis del error.

La metodología para el modelado de las distribuciones a posteriori de los parámetros sísmicos se obtiene de los siguientes pasos:

1. Seleccionar el catálogo (con o sin replicas).

2. Clasificar la información con base en la región de interés.

3. Seleccionar el modelo de distribución (normal o exponencial).

4. Calcular la distribucion a posteriori con el método de Monte Carlo.

5. Calcular la divergencia KUllback-Leibler entre distribuciones.

Cálculo de distribuciones a posteriori para parámetros sísmicos en una región de Italia para catálogos de información que han sido procesado con técnicas de rechazo de réplicas.

La metodología inferencia Bayesiana por el método Monte Carlo se aplicó a tres catálogos con la información sísmica de la región de interés en Italia, los datos proporcionados en estos catálogos son resultados de metodologías de rechazo de réplicas: Método de Reasenberg en el cual se identifican réplicas de acuerdo con la zona y el tiempo de interacción, Método de Knopoff y Gardner con ventana de Grunthal y Método de Knopoff y Gardner con ventana de Humhammer, en los casos de los métodos con ventanas se analiza un evento con magnitud M y los eventos que se encuentren en un tiempo cercano t (Para más detalles sobre la aplicación de los métodos revisar las referencias [78], así como las referencias en ese mismo trabajo). Los catálogos fueron obtenidos por medio del proyecto *CNR-CINVESTAV del Consiglio Nazionale delle Ricerche CNR.*

En la figura 5.3 se observan los resultados del modelado del error entre las distribuciones modeladas en nats para los parámetros de magnitud, distancia y tiempo con el modelo exponencial cuando se aplica el método de rechazo de perturbaciones de Rasenberg en relación con el número de muestras utilizadas en el método de Monte Carlo la línea punteada indica un error menos o igual al 10 por ciento.

En la figura 5.4 se muestran los resultados del error de modelación para la distribución e nats de los parámetros sísmicos en Italia cuando se le aplica el método de rechazo de replicas con la ventana de Grunthal. Estas distribuciones emplean el modelo exponencial. La línea punteada indica un error menor o igual a 10 por ciento.

Por último, se observa el error de distribución en nats de los parámetros sísmicos de Italia cunado se aplica el método de rechazo de replicas con la ventana de Humhammer en la figura 5.5. En estos resultados se aplicó la forma normal para obtener las distribuciones de los parámetros sísmicos de Italia y la linea punteada indica cuando se obtiene un error menor o igual a 10 por ciento.

Como se puede observar de los resultados, la predicción de las distribuciones depende del número de muestras que se seleccionan en cada caso, sin embargo, en ambos modelos de distribución se ha producido satisfactoriamente la predicción de la distribución de probabilidad de los parámetros sísmicos para esta región de Italia. Estos resultados otorgan información sobre probabilidades de los parámetros de magnitud, distancia y tiempo de

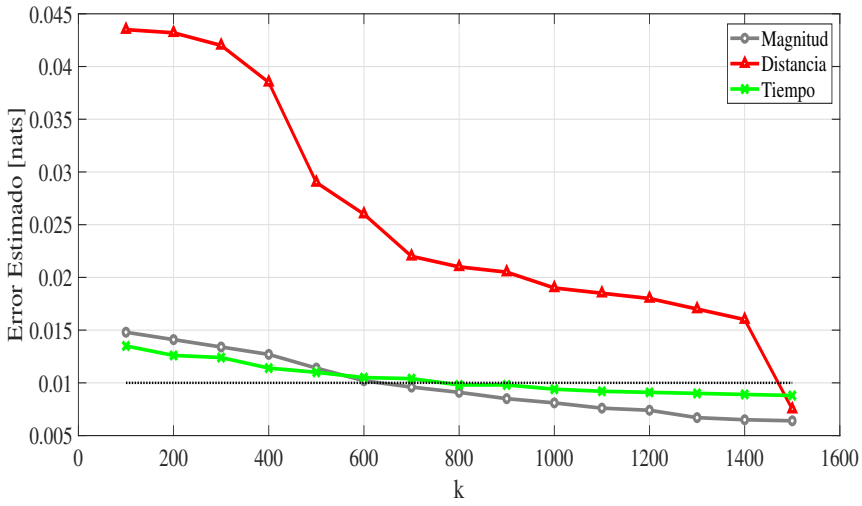


Figura 5.3: Error de modelado de los parámetros sísmicos en el catálogo con el método de Rasenberg.

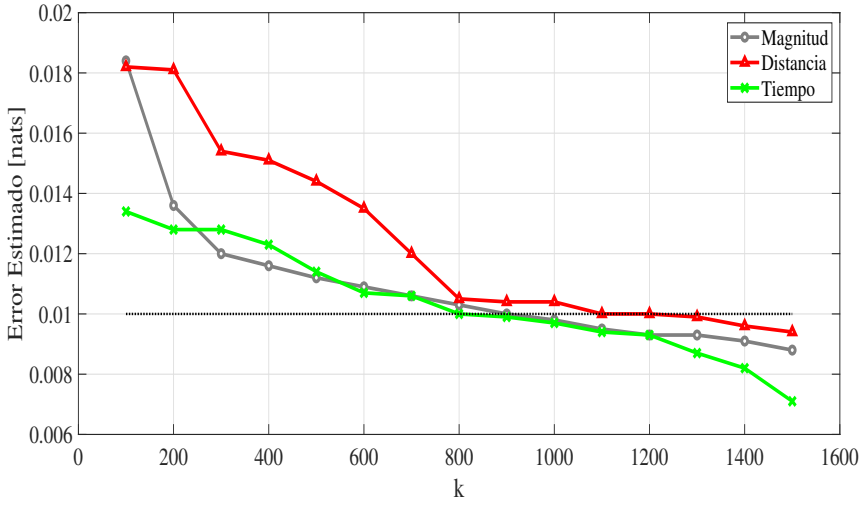


Figura 5.4: Error de modelado de los parámetros sísmicos en el catálogo con el método de la ventana de Grunthal.

los sismos con lo cual podemos obtener un conocimiento de los parámetros para eventos próximos.

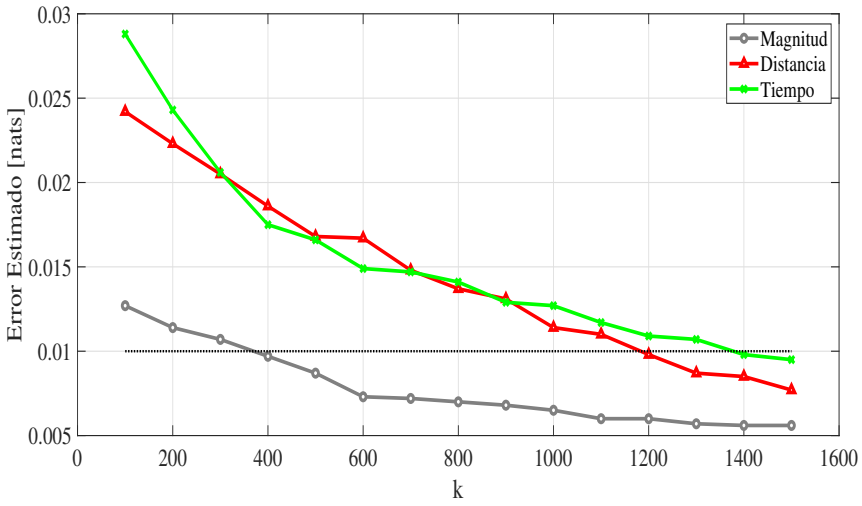


Figura 5.5: Error de modelado de los parámetros sísmicos en el catálogo con el método de la ventana de Uhmhammer.

5.1.1. Cálculo de las distribuciones a posteriori usando datos de información reciente aplicado a parámetros sísmicos.

Con la misma información sísmica de los catálogos de Italia se aplicó la metodología de inferencia Bayesiana por medio la aplicación de datos con información reciente de los parámetros de magnitud, distancia entre eventos y tiempo entre eventos de los sismos.

En el cálculo de las distribuciones posteriores por medio de la información reciente como se observa en la figura 3.3 se realiza mediante los siguientes pasos:

1. Dados los datos históricos X_1 calcular la distribución de probabilidad por medio del método de Monte Carlo.

2. Obtener $p(\theta \mid X_1)$ y generar el nuevo conocimiento a priori $p(\theta \mid X_2)$.

3. Para los datos recientes X_2 iniciar el método de Monte Carlo para la distribución a posteriori.

4. Calcular la nueva distribución a posteriori $p(\theta \mid X_1, X_2)$.

5. Calcular la divergencia KUllback-Leibler entre distribuciones.

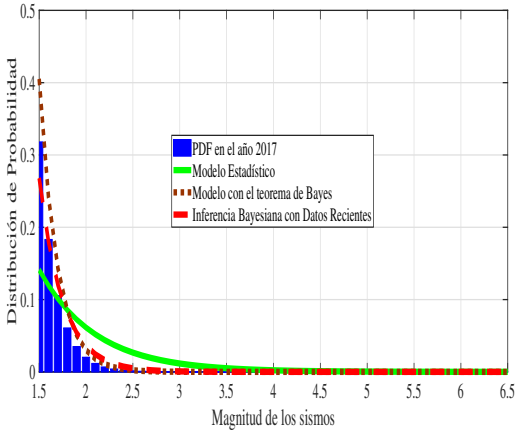
En las siguientes figuras 5.6 se observan los modelos de distribución obtenidos para los parámetros de magnitud, distancia y tiempo generados desde el catálogo sin técnica de rechazo de réplicas de la región de Italia. En el método se utilizó la información de los años 2014 a 2015 como datos históricos X_1, mientras que los datos que se tienen en el año 2016 se usan como información reciente X_2 en la inferencia Bayesiana para predecir la distribución que se tiene en el año 2017.

El procedimiento de inferencia Bayesiana con datos recientes se aplicó para los 3 catálogos que han sido procesados con las técnicas de rechazo de réplicas de los sismos. Se puede observar los resultados de las distribuciones a posteriori de los parámetros sísmicos analizando la divergencia de las distribuciones en nats para cada uno de los catálogos en la tabla 5.1.

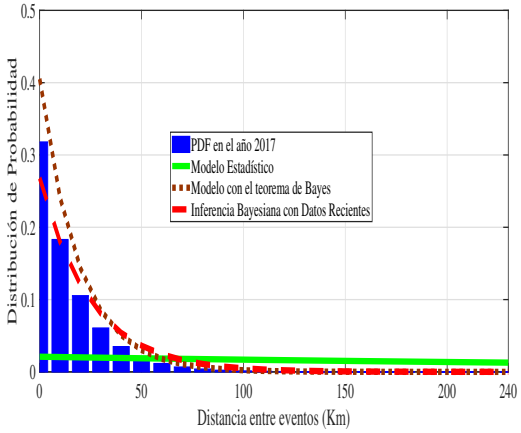
Tabla 5.1: Errores de predicción.

Catalogo	Distribución	Modelo Estadístico $\times 10^{-3}$	Método Bayesiano $\times 10^{-3}$	Inferencia Bayesiana $\times 10^{-3}$
ITcat	Magnitud	17.9	8.9	4.5
19952018zH	Distancia	56.5	10.5	8.2
60Km	Tiempo	9.7	35.8	5.1
IT_dec	Magnitud	12.9	4.9	3.3
_RS_z	Distancia	35.6	10.5	7.7
	Tiempo	15.2	11.0	9.7
IT_dec	Magnitud	12.9	4.9	3.2
_GR_z	Distancia	48.4	9.3	4.5
	Tiempo	75	24.4	12.7
IT_dec	Magnitud	13.0	50.0	3.0
_UH_z	Distancia	19.5	19.8	2.8
	Tiempo	18.0	9.5	5.6

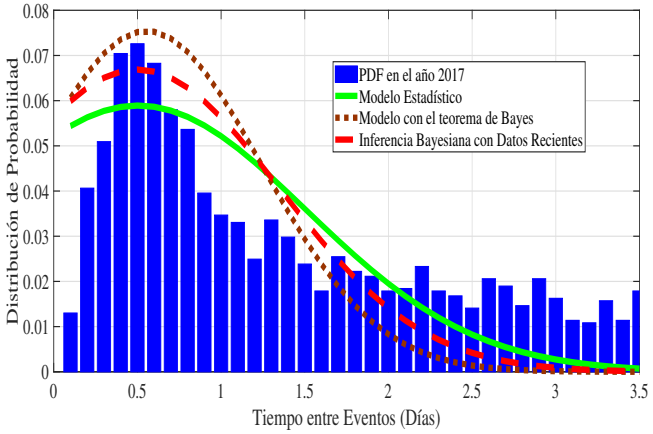
En la tabla 5.1 ITcat19952018zH60km representa el catálogo de información sísmica de Italia que no ha sido procesado con algún método de rechazo de réplicas, $IT\_dec\_RS\_z$ es el catálogo con el método de Rasenberg, $IT\_dec\_GR\_z$ representa el catálogo con el método ventana de Grunthal y $IT\_dec\_UH\_z$ es el método con la ventana de Umerhammer. Además de que se puede observar una comparación de los errores obtenidos



(a) Magnitud



(b) Distancia entre eventos



(c) Tiempo entre eventos

Figura 5.6: Distribuciones Posteriores.

por la divergencia entre las distribuciones utilizando el modelo estadístico para una distribución a posteriori como se observa en la figura 5.7y utilizando el teorema de Bayes aplicando la metodología de Monte Carlo para distribuciones a posteriori como se observa en la figura 5.8. Obteniendo los valores de las divergencias en nats.

El cálculo de distribuciones a posteriori por medio de información reciente ha conseguido pronosticar la distribución de probabilidad de los parámetros sísmicos en los catálogos de

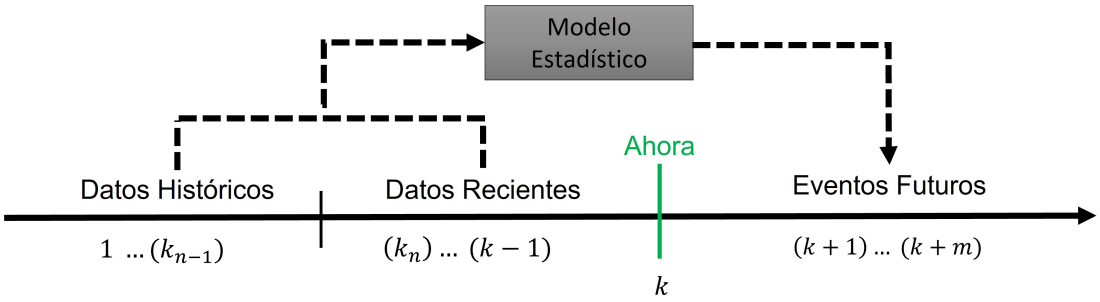


Figura 5.7: Modelo estadístico para distribuciones de probabilidad.

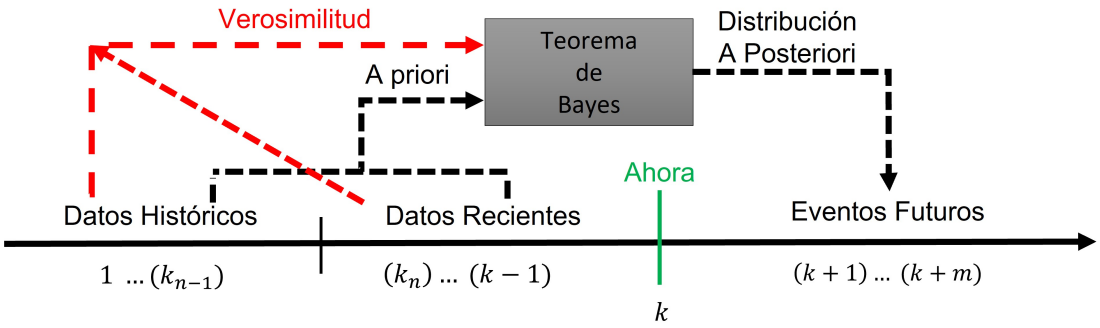


Figura 5.8: Teorema de Bayes para distribución a posteriori.

información de Italia obteniendo mejoras en el modelo de la distribución a posteriori como se observa en la tabla 5.1 en comparación con la aplicación del modelo estadístico y el teorema de Bayes para las distribuciones a posteriori. El resultado de distribuciones de probabilidad a posteriori tiene una mejor actualización cuando se utiliza información reciente acercándonos a las predicciones de probabilidad para los parámetros sísmicos.

5.2. Cálculo de distribuciones de a posteriori de parámetros sísmicos mediante una red neuronal y la inferencia Bayesiana.

Se calculan las distribuciones a posteriori de los parámetros sísmicos en el país de México utilizando la inferencia Bayesiana y una red neuronal con lo cual encontramos información de la probabilidad de estos parámetros en eventos próximos. Para esta

aplicación se ha seleccionado información sísmica que se presenta en el país de México, basado en un catálogo de información que tiene información recabada desde el primero de enero del 1985 hasta el 30 de septiembre del 2020 que se encuentra disponible en `http://www2.ssn.unam.mx:8080/catalogo/`. Tomando la información de los parámetros de magnitud y distancia entre eventos para realizar el modelo de la distribución a posteriori utilizando la información de los años 2015 y 2016 para construir el modelo a priori y los datos entre los años 2017 y 2018 para la aplicación de la red neuronal *MDN* para obtener una distribución de probabilidad la cual se actualiza la distribución a priori para obtener la distribución a posteriori del modelo en los años 2019 y 2020.Para el calculo de las distribuciones a posteriori utilizando una red neuronal *MDN* en la inferencia Bayesiana se utiliza el modelo de distribución normal cumpliendo con el teorema de limite central y con el modelo utilizado en la red neuronal.

1. Generar la información a priori $p(\theta)$.

2. Con y calcular $p(\hat{y})$ con la NN usando datos recientes.

3. Usar $p(\hat{y})$ como $p(x|\theta)$ para actualizar $p(\theta)$.

4. Obtener la distribucion a posteriori $p(\theta|X)$.

5. Analizar divergencia Kullback-Leibler.

En las figuras 5.9 y 5.10 se observa los resultados de las distribuciones a posteriori de parámetros sísmicos de magnitud y distancia entre eventos de México cuando se aplica la inferencia Bayesiana por medio de una red neuronal y una comparación con el método de Monte Carlo para distribuciones a posteriori.

En la tabla 5.2 se puede observar los resultados para la distribución a posteriori usando una red neuronal en el procedimiento de inferencia Bayesiana comparada con el método de Monte Carlo para distribuciones a posteriori, usando la divergencia de Kullback-Liebler para determinar el error de modelado estimado en nats.

Los resultados en la 5.2 muestran que la aplicación de redes neuronales en la inferencia Bayesiana puede obtener resultados para distribuciones a posteriori para la magnitud y la distancia entre eventos de sismos en México. La aplicación de la red en la inferencia Bayesiana para el cálculo de distribuciones a posteriori muestran una mejor actualización como se obtuvo en la inferencia Bayesiana con datos recientes.

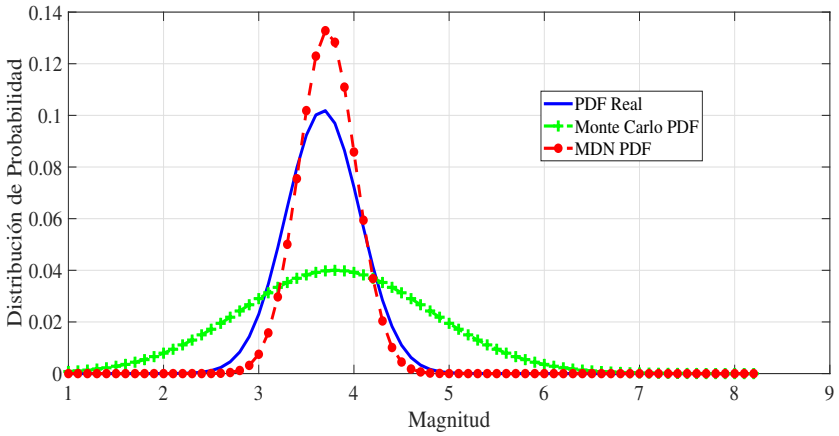


Figura 5.9: Distribucion posterior para magnitud de sismos en México.

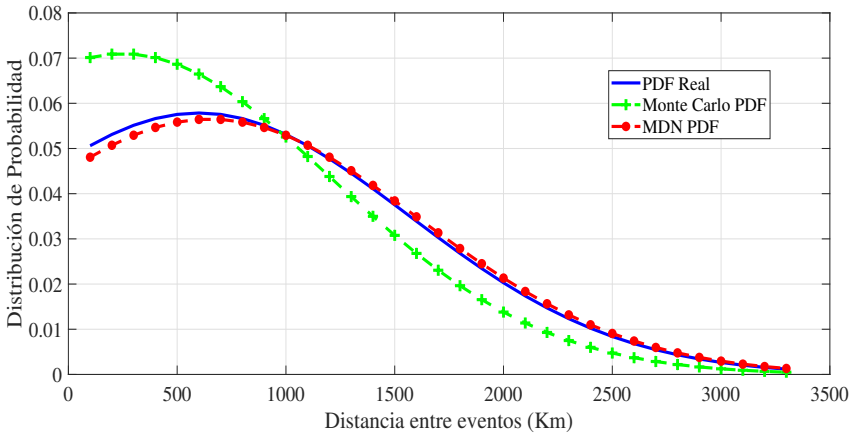


Figura 5.10: Distribucion posterior para distancia de sismos en México.

Calculo de las distribuciones posteriores para la magnitud de sismos en Italia y México con Redes Neuronales.

Se implementó el modelado de distribución a posteriori por medio de una red neuronal *MDN* que esta entrenada por los métodos de aprendizaje obteniendo el modelo de distribución como en la sección 4.1.1 y el método de Mínimos Cuadrados (LS) para la optimización de los pesos en la estructura de la red como en la sección 4.1.2, con el objetivo de comparar el rendimiento de las redes neuronales en la aplicación de las distribuciones posteriores utilizando la información en los años 2015 y 2016 como un conocimiento a prior

Tabla 5.2: Error de modelado en información sísmica ($\times 10^{-3}$)

	Monte Carlo	Inferencia Bayesiana con MDN
Magnitud	26.5	2.2
Distancia	19.0	3.9

en la metodología y para la identificación de los datos por medio de la red se utilizan los datos en los años 2017 y 2018.

En la figura 5.11 se muestran los resultados para el modelado de la distribución a posteriori de la magnitud de los sismos en Italia, mientras que en la figura 5.12 se muestra el resultado de la distribución a posteriori de la magnitud de los sismos en México. Los resultados de las distribuciones a posteriori en las figuras se obtuvieron mediante el método de Monte Carlo (MC), y la inferencia Bayesiana con una red neuronal MDN con dos métodos del entrenamiento. A la combinación de la inferencia Bayesiana y la red MDN se nombra Bayes-MDN en las figuras 5.11 y 5.12.

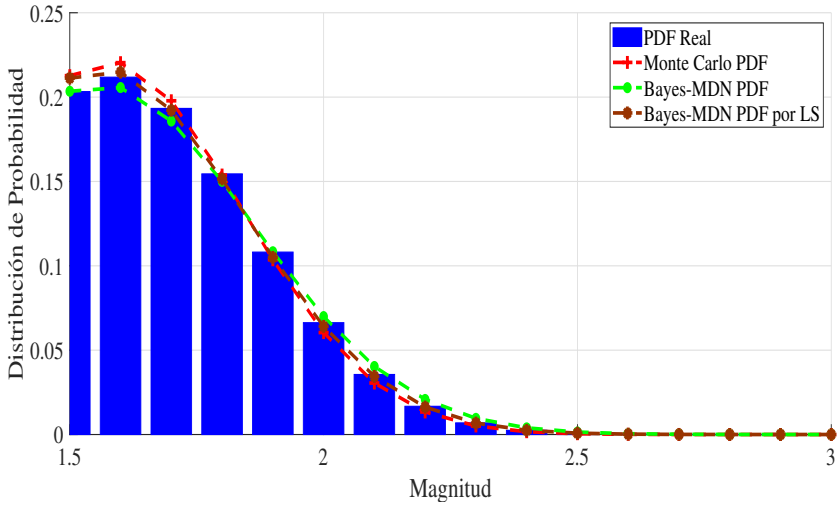


Figura 5.11: Distribución posterior de la magnitud en Italia.

La Tabla 5.3 muestra los resultados de la divergencia en nats para el modelado de la distribución a posteriori de la información sísmica obtenido de la aplicación de la inferencia

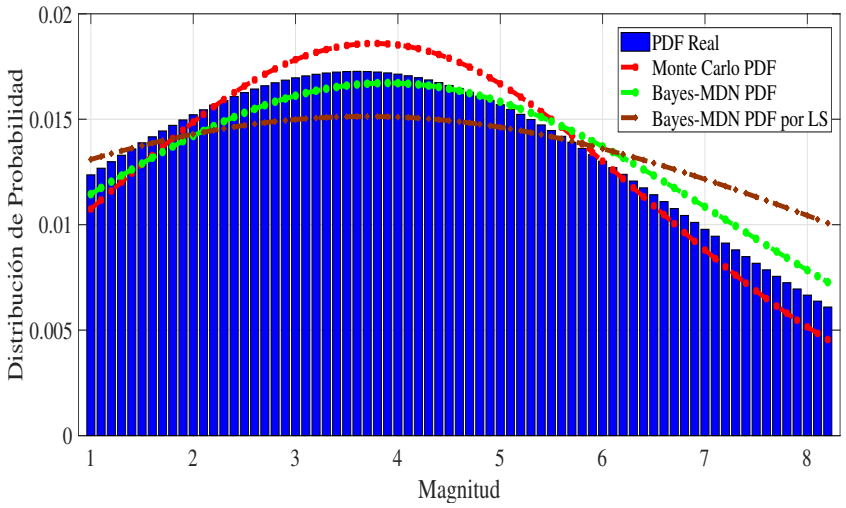


Figura 5.12: Distribución posterior de la magnitud en México.

por medio de una red neuronal *MDN*, comparando con el método de Monte Carlo *MC* y la inferencia Bayesiana con la red neuronal entrenada por medio del método de Mínimos Cuadrados (LS).

Tabla 5.3: Errores de modelado de distribucion posterior ($\times 10^{-3}$)

	Error de MC	Error de MDN	Error de Mínimos Cuadrados
Magnitud de Italia	2.1	0.9	0.6
Magnitud de México	3	0.97	0.23

5.3. Identificación de series de tiempo usando el método de ajuste fino de los pesos.

Se han seleccionado dos sistemas no lineales para analizar la identificación de sistemas dinámicos por medio de redes neuronales y el ajuste fino de los pesos por medio de la información estadística $p(y|x)$. Aplicando una red neuronal de dos capas para realizar la identificación del sistema dinámico en un primer lote de entrenamiento y con la misma estructura del número de pesos y número de capas se realiza el ajuste fino de los pesos en la red con un segundo lote de entrenamiento por medio de la información estadística.

Horno de gas.

Este sistema cuenta con 296 puntos de información disponible de su entrada u y salida y con los cuales se puede realizar la identificación por medio de la red neuronal. La entrada $u(k)$ representa el flujo de metano que entra hacia el horno, mientras que la salida $y(k)$ representa el dióxido de carbono en la salida del mismo horno, obteniendo estas medidas con un intervalo de 9 segundos [18]. Para la aplicación de la red neuronal se han tomado los primeros 149 datos del sistema para realizar la identificación y los restantes 146 datos que van desde el dato 150 al 296 para prueba de la red entrenada.

En la figura 5.13 se tiene los resultados de la identificación del sistema de horno de gas por medio de la red neuronal en cada etapa de entrenamiento, siendo la primera la identificación por red neuronal y segunda por ajuste fino de los pesos. En este caso se realizó un recorte en el número de pesos de la red neuronal empezando con una estructura NN_0 con 30 pesos en la capa de salida y 10×30 pesos en la capa oculta, realizando el recorte de la estructura $\bar{\xi} = 12$ veces encontrando la mejor divergencia KL en la poda número $\bar{\xi} = 9$ seleccionando la estructura de la red NN_9 la cual resultó con 10×7 pesos en la capa oculta y 7 pesos en la capa de salida.

Sistema de tanques en cascada.

Para el sistema de tanques conectados en cascada se tienen 1024 datos de información disponibles que representa el sistema de control de nivel en dos tanques con alimentación independiente por medio de dos bombas. El sistema está principalmente modelado por los principios de Bernoulli y de conservación de la masa, pero el modelo no considera el efecto de sobre flujo que pueda aparecer [63]:

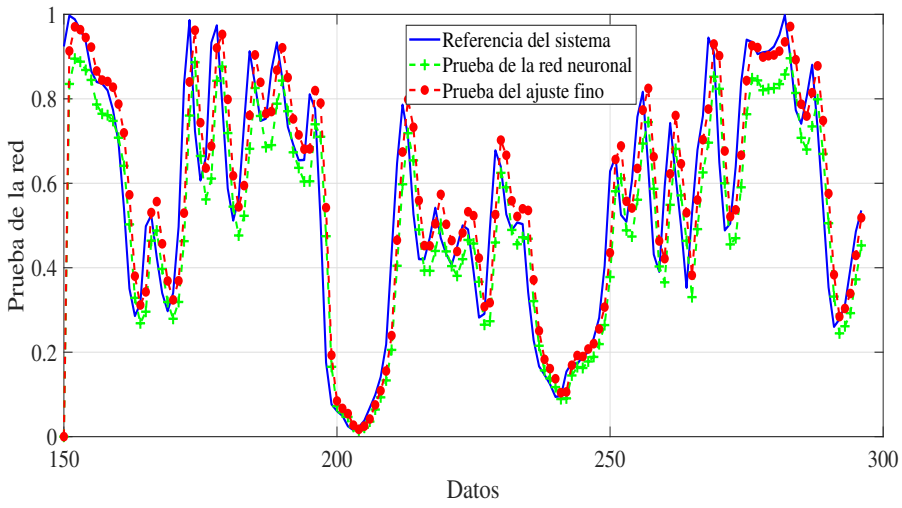


Figura 5.13: Ajuste fino en el modelado del sistema de gas.

$$\dot{x}_1(t) = -k_1\sqrt{x_1(t)} + k_4 u(t) + \zeta_1(t), \tag{5.2}$$

$$\dot{x}_2(t) = k_2\sqrt{x_1(t)} - k_3\sqrt{x_2(t)} + \zeta_2(t), \tag{5.3}$$

$$y(t) = x_2 + \xi(t) \tag{5.4}$$

donde $u(t)$ representa la señal de entrada, $x_1(t)$ y $x_2(t)$ son los estados del sistema, $\zeta(t)$, $\zeta(t)$ y $\xi(t)$ son fuentes de ruido aditivas, por último k_1, k_2, k_3 y k_4 representan constantes que dependen del modelado en el sistema [63].

En la figura 5.14 se observa la identificación del sistema de tanques conectados en casacas en la etapa de prueba de la red neuronal, así como la etapa de prueba del ajuste fino de los pesos cuando la red mantiene la misma estructura de dos capas sin poda utilizando 600 datos de la entrada y salida del sistema para el entrenamiento de la red y a partir del dato 601 hasta el dato 1024 se ha utilizado como referencia para probar la red neuronal entrenada en cada lote. Utilizando una estructura inicial NN_0 con 20 pesos en la capa de salida, con 6×10 pesos en la capa oculta y se realizó la poda de la estructura $\bar{\xi} = 10$ veces encontrando la mejor divergencia KL en el paso de la poda $\bar{\xi} = 4$ seleccionando la estructura NN_4 que resultó con 4 pesos en la capa salida y 6×4 pesos en la capa oculta.

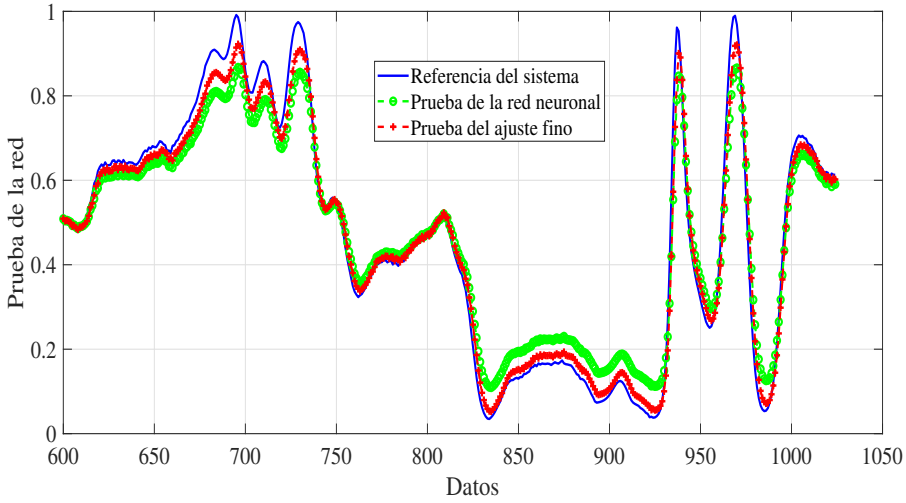


Figura 5.14: Ajuste fino en el modelado del sistema en cascada

En la tabla 5.4 se tiene un análisis del error acumulado en la red analizado mediante el error cuadrático medio (MSE) en la etapa de prueba de la red y la etapa de prueba en la red mediante el ajuste fino de los pesos.

Tabla 5.4: Errores modelados por medio de MSE $\times 10^{-3}$

Ajuste fino		
	Red neuronal	Ajuste fino
Horno de Gas	9.0	3.8
Sistema en Cascada	3.3	0.97

5.3.1. Identificación de sistemas dinámicos comparando resultados del entrenamiento BP y ELM en el primer lote de entrenamiento.

Se aplicaron los tipos de entrenamiento BP y ELM en el primer lote de entrenamiento de la red neuronal antes de realizar el ajuste fino de los pesos en la red, para obtener la identificación de los sistemas dinámicos del horno de gas y el sistema de tanques conectados en casada. Con estos resultados se busca comparar el rendimiento de los entrenamientos en

la red, así como la aplicación y rendimiento del ajuste fino de los pesos en la estructura de la red una vez entrenada en la primera etapa. Mediante los siguientes pasos se realiza el proceso de ajuste fino de los pesos en la red neuronal para la identificación de los sistemas dinámicos.

1. Entrenar el primer lote con BP o ELM.

2. Guardar los pesos W, V.

3. Entrenar segundo lote con información estadística $p(y|x)$.

4. Obtener los nuevos pesos W, V.

5. Probar la red con el ajuste fino de los pesos con los datos de prueba y de la salida del sistema.

Horno de gas BP vs ELM

Del total de 296 puntos de análisis contenidos en el catálogo se utilizan 50 puntos de información de la entrada y la salida del sistema para el entrenamiento de la red mientras que los 246 puntos restantes se han utilizado como información desconocida para probar la red entrenada en cada una de las etapas de entrenamiento para el proceso de ajuste fino de los pesos de la red neuronal.

Los resultados del ajuste fino de los pesos en la red en el sistema de horno de gas se muestran en la Figura 5.15. En este caso se realizó un recorte en el número de pesos de la red neuronal empezando con una estructura NN_0 con 30 pesos en la capa de salida y 10×30 pesos en la capa oculta, realizando el recorte de la estructura $\bar{\xi} = 12$ veces encontrando la mejor divergencia KL en la poda número $\bar{\xi} = 9$ seleccionando la estructura de la red NN_9 la cual resultó con 10×7 pesos en la capa oculta y 7 pesos en la capa de salida, comparando los resultados que se obtienen de identificar el sistema cuando se entrena a la red neuronal en el primer lote de entrenamiento con los métodos de propagación hacia atrás del error y el Aprendizaje Extremo en la primera etapa de la red neuronal.

Sistema de tanques en cascada BP vs ELM

En este caso retomamos los 1024 puntos de información que se tienen en el sistema de dos tanques conectados en cascada [63], con el objetivo de comparar los métodos de entrenamiento que se ocupan en la primera etapa de entrenamiento para la metodología de

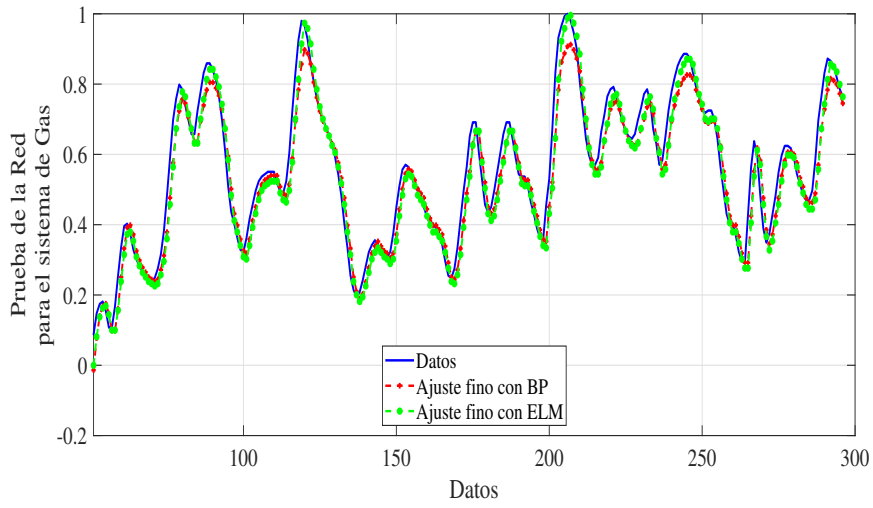


Figura 5.15: Prueba de la red en la etapa de ajuste fino de los pesos para el sistema de horno de gas.

ajuste fino de los pesos. Los resultados del modelado de las redes neuronales se muestran en la Figura 5.16. Para los resultados de la identificación de este sistema se utilizan 300 puntos de datos para entrenar la red neuronal mediante los métodos de entrenamiento de propagación hacia atrás del error y Aprendizaje Extremo, mientras que el resto de los datos de la referencia se utilizan como puntos de prueba para la propia red. Utilizando una estructura inicial NN_0 con 20 pesos en la capa de salida, con 6×10 pesos en la capa oculta y se realizó la poda de la estructura $\bar{\xi} = 10$ veces encontrando la mejor divergencia KL en el paso de la poda $\bar{\xi} = 4$ seleccionando la estructura NN_4 que resultó con 4 pesos en la capa salida y 6×4 pesos en la capa oculta

Debido a que los resultados son muy cercanos en el momento de modelar al sistema se opta por generar un pequeño acercamiento (*zoom*) de los últimos datos para observar las diferencias entre los métodos al aplicar el ajuste finos de los pesos en la red neuronal por cada uno de los tipos de aprendizaje aplicados. El acercamiento de los resultados de la prueba de la red neuronal en la etapa de ajuste fino se observa en la figura 5.17.

En la tabla 5.5 se observan los resultados del erro cuadrático medio (MSE) en la etapa de prueba para el procedimiento de ajuste fino de los pesos en la red neuronal usando la información con estadística $p(y|x)$ comparando cada uno de los métodos de aprendizaje BP y ELM que se aplica en la red neuronal en wl primer lote de entrenamiento de esta metodología.

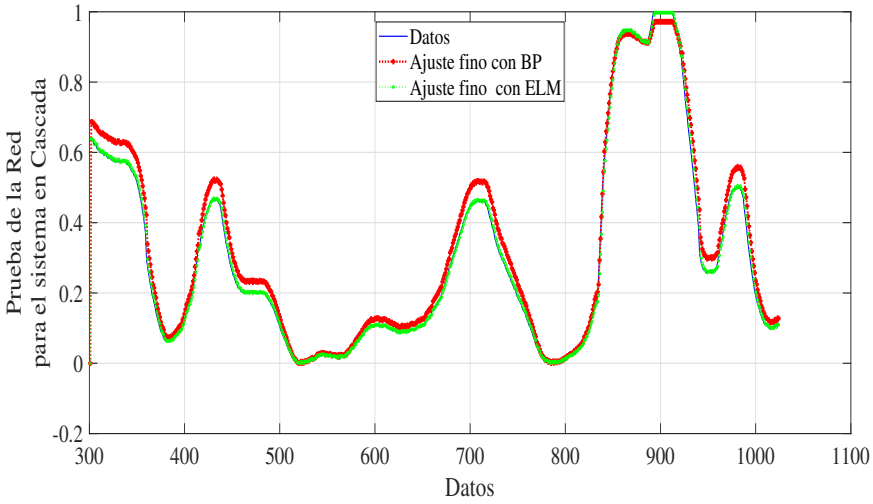


Figura 5.16: Prueba de la red en la etapa de ajuste fino de los pesos para el sistema tanques en cascada.

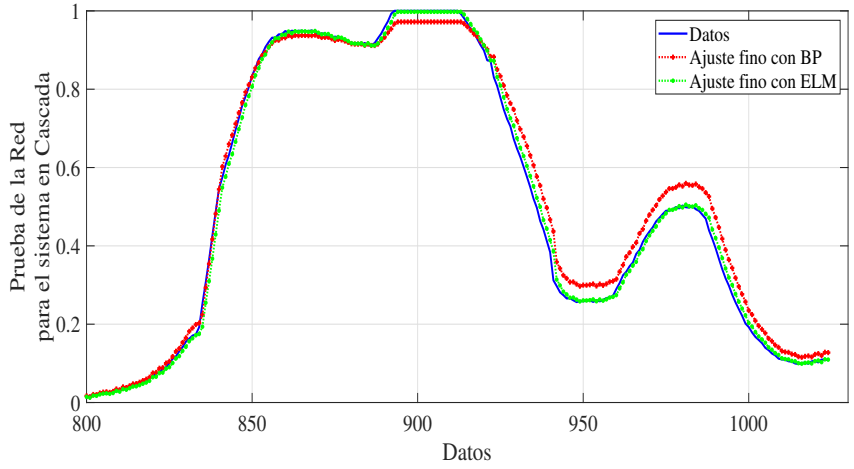


Figura 5.17: Acercamiento en el modelado con ajuste fino en la red para el sistema de tanques en cascada

Tabla 5.5: Rendimiento del ajuste fino en la red neuronal

	Sistema de Gas		Sistema en Cascada	
	Error en la NN	Error en el Ajuste fino	Error en la NN	Error en el ajuste fino
BP	1.5	0.8	0.6	0.05
ELM	1.4	0.17	0.3	0.006

5.3.2. Cálculo de la distribución en la dinámica de los sistemas dinámicos por medio de una red MDN.

Para este sistema también se realizó el cálculo de la distribución de probabilidad que se tiene en la salida del sistema para modelar por medio de la red neuronal *MDN* comparando los resultados que se obtienen en la red cuando se aplican los procedimientos de aprendizaje BP y Mínimos Cuadrados (LS), así como una comparación con el método de Monte Carlo para obtener la distribución de probabilidad del sistema analizado. Estos resultados se pueden observar en la figura 5.18.

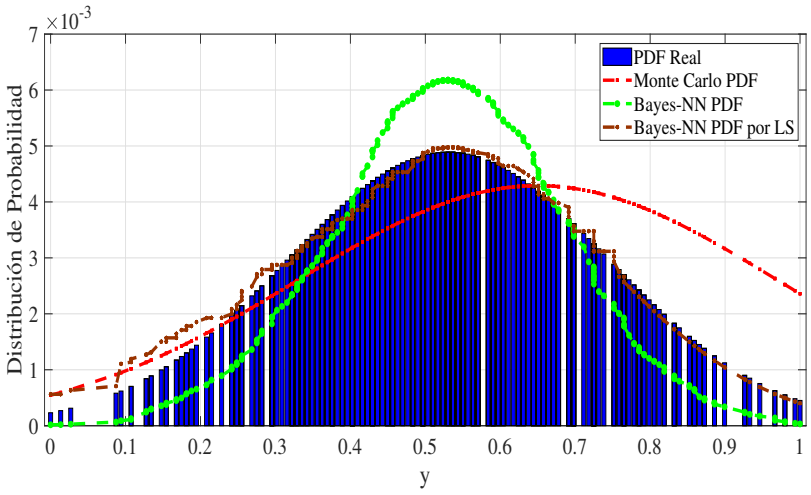


Figura 5.18: Distribución de la salida del sistema de horma de gas.

La figura 5.19 muestra la distribución de probabilidad en la salida del sistema de tanques conectados en cascada, donde las distribuciones se obtienen mediante el modelado Bayesiano de Monte Carlo y las redes neuronales *MDN* utilizando los métodos de entrenamiento de propagación hacia atrás (BP) y el entrenamiento por Mínimos Cuadrados (LS), utilizando la información encontrada entre los puntos 300 a 600 de la información disponible en la

salida y del sistema.

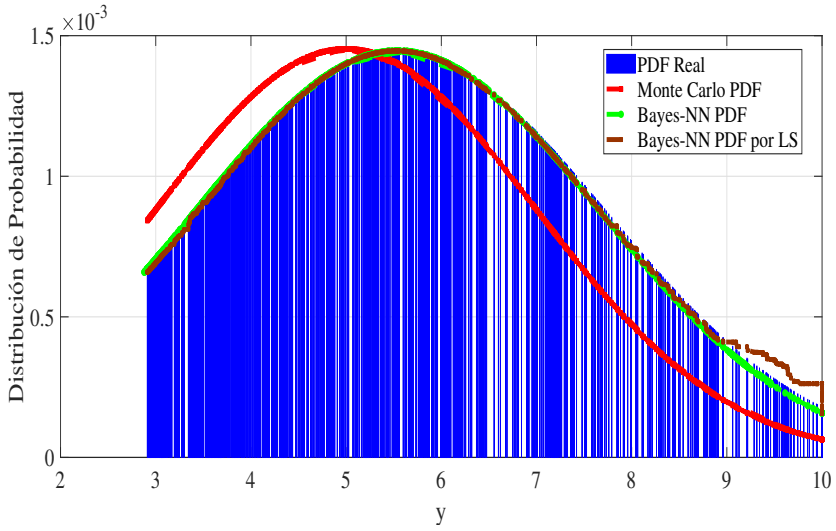


Figura 5.19: Distribución de la salida del sistema en Cascada.

Los resultados de la divergencia de las distribución de probabilidad en la salida de los sistemas dinámicos por medio de la red neuronal *MDN* se puede observar en la tabla 5.6 calculados por medio de la divergencia de Kullback-Leibler en nats, comparando los resultados obtenidos con el método de Monte Carlo *MC* y el procedimiento con la red neuronal para inferencia Bayesiana utilizando los métodos de entrenamiento BP y Mínimos cuadrados (LS) en la red neuronal.

Tabla 5.6: Red neuronal con inferencia Bayesiana ($\times 10^{-3}$).

	Sistema de Gas		Sistema en Cascada	
	Error en la Red	Divergencia KL	Error en la Red	Divergencia KL
BP	2.4	1.2	0.6	0.53
ELM	1.4	1.1	0.3	0.71
MC	-	2.07	-	1.28

Análisis de los tiempos de cómputo en la aplicación de la metodología de ajuste fino de los pesos para el modelado de los sistemas dinámicos.

En el proceso de identificación de sistemas dinámicos se puede analizar mediante el error de modelado que se tiene en el proceso de la red neuronal, ya que se puede observar la dinámica del error cuando la red neuronal es puesta a prueba. Con lo que se puede observar el error de modelado de la red neuronal a lo largo de los puntos de prueba analizados para cada sistema.

En la figura 5.20 se observa el error modelado en la etapa de prueba de la red neuronal para el sistema de horno de Gas y el error modelado en la etapa de prueba cuando se realiza el ajuste fino de los pesos en la red.

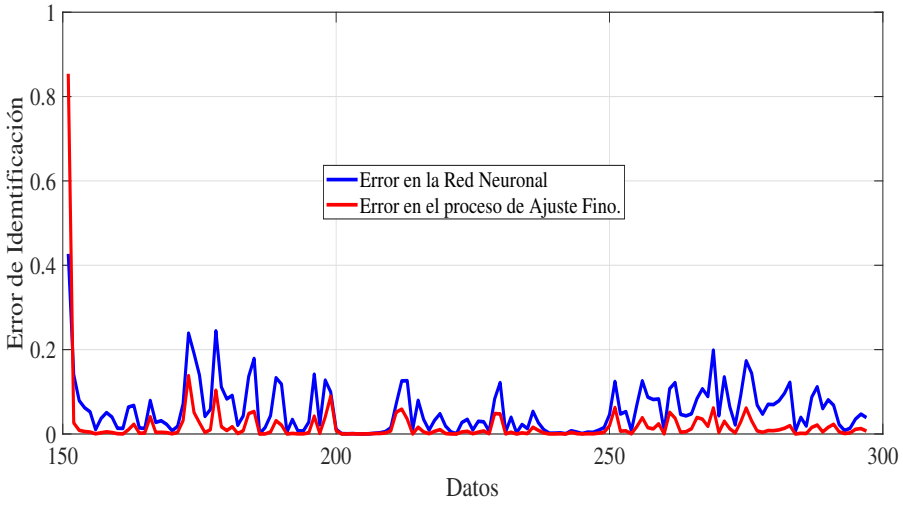


Figura 5.20: Error de identificación para el sistema de Gas.

El análisis de la dinámica del error en la etapa de prueba de la red y etapa de prueba en el ajuste fino de los pesos se realizó para obtener el modelo del sistema de tanques conectados en cascada como se puede observar en la figura 5.21.

Se toma el análisis del tiempo que le toma la aplicación del programa para obtener el modelo de los sistemas dinámicos en la etapa de prueba de la red neuronal con el objetivo de analizar si existe un costo computacional importante al momento de la ejecución de los procedimientos de identificación del sistema dinámico y la identificación del sistema

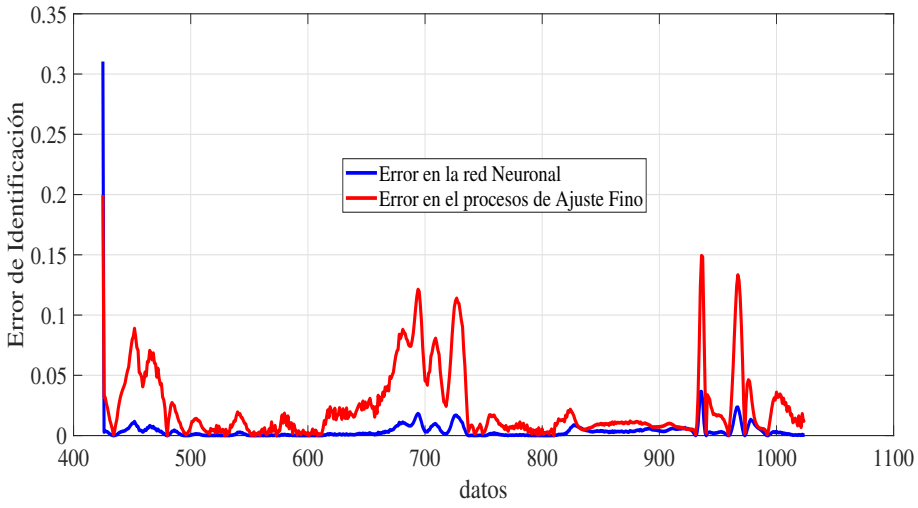


Figura 5.21: Error de identificación para el sistema de Tanques en Cascada.

por medio del ajuste fino de los pesos. Tomando en cuenta que se tienen las mismas características técnicas de la computadora en todo momento: con un procesador $x64$ de 64 bits, con memoria RAM de 4 GB y un almacenamiento de tipo SSD de un Terabyte de capacidad, apoyándose de la herramienta de Matlab 2019a, siendo la única tarea ejecutada en ese momento.

En la tabla 5.3 se muestran los resultados del tiempo para la ejecución de la etapa de prueba en la red neuronal y una comparación con el tiempo que tarda la aplicación del ajuste fino de los pesos en la red neuronal midiendo el tiempo en segundos (s).

Tabla 5.7: Tiempo de ejecución de las redes neuronales.

	Perceptron	Ajuste-fino
Sistema de Gas	0.025071s	0.106857s
Sistema en Cascada	0.033046s	0.11829s

Capítulo 6

Conclusiones

La inferencia Bayesiana nos permite encontrar distribuciones de probabilidad con información a posteriori con base en creencias que se obtienen de la información analizada, en este trabajo se aplicó a los parámetros sísmicos generando un pronóstico de la distribución de probabilidad en los parámetros de magnitud eventos, distancia entre eventos y tiempo entre eventos, además se tuvieron mejoras en el cálculo de las distribuciones a posteriori mediante el uso de información reciente. Con estas metodologías nos acercamos a poder predecir la probabilidad de eventos futuros.

Las metodologías aplicadas en la inferencia Bayesiana por medio de la actualización de las distribuciones con información reciente obtuvieron mejores resultados en el cálculo de las distribuciones a posteriori, siguiendo esta idea se aplicó una red neuronal para modelar una distribución de probabilidad la cual se aplica a la inferencia Bayesiana para calcular la distribución a posteriori. De esta aplicación se obtuvieron buenos resultados al calcular la distribución a posteriori de los parámetros sísmicos comparando con los métodos de Monte Carlo.

Se pudo generar una estructura basada en redes neuronales para la identificación de sistemas dinámicos agregando la información estadística de los datos de entrada y salida del sistema. La cual se entrenó en dos lotes donde el primer lote actualiza los pesos con la información de entrada y salida mientras que el segundo lote obtuvo un ajuste fino de los pesos utilizando información estadística de los datos de entrada y salida. La aplicación del ajuste fino de los pesos en la red neuronal generó mejoras de rendimiento en la identificación sistemas dinámicos comparando con una red neuronal de estructura simple.

Capítulo 7

Trabajo futuro

Se pueden implementar inferencia Bayesiana para distribuciones de probabilidad combinado con las redes neuronales que tengan modelos de distribución distintos al modelo normal. Obteniendo resultados de distribuciones posteriores con cualquier tipo de modelo de distribución que pueda aparecer en la información analizada.

El cálculo de distribuciones a posteriori se puede implementar en problemas donde la información cambia constantemente y toma mayor relevancia la información reciente como en el caso eventos sísmicos, datos de contaminación o a los valores económicos en el mercado de bolsa.

La identificación del sistema dinámico que se obtuvo mediante la metodología presentada en este trabajo puede servir para realizar el control para el sistema analizado.

Bibliografía

[1] Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. *A learning algorithm for Boltzmann machines*. Cognitive science, 1985, vol. 9, no 1, p. 147-169.

[2] Åström, K. J., & Murray, R. M. *Feedback systems*. Princeton university press, 2010.

[3] Alonso, Diego & Tubau, Elisabet. *Inferencias bayesianas: una revisión teórica*. Anuario de Psicología, 2002, vol. 33, no 1, p. 25-47.

[4] Ben-gal, I. *Bayesian networks*. Encyclopedia of statistics in quality and reliability, 2008, vol. 1.

[5] Ben-Gal, Irad, et al. *Identification of transcription factor binding sites with variable-order Bayesian networks*. Bioinformatics, 2005, vol. 21, no 11, p. 2657-2666.

[6] Bernardo, J., Berger, J., Dawid, A. P. A. F. M. S., & Smith, A. *Regression and classification using Gaussian process priors*. Bayesian statistics, 6, 475. 1998.

[7] Bishop, Christopher M. *Mixture density networks*. 1994.

[8] Bouvrie, J. *Notes on convolutional neural networks*. 2006.

[9] Blalock, Davis, et al. *What is the state of neural network pruning?*. arXiv preprint arXiv:2003.03033, 2020.

[10] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[11] Briega, R. E. *Matemáticas, análisis de datos y python*. 2018.

[12] Cornell, C. A. *Bayesian statistical decision theory and reliability-based design*. En International Conference on Structural Safety and Reliability. Pergamon, 1972. p. 47-68.

[13] Cortes, C., & Vapnik, V. *Support-vector networks.* Machine learning, 1995, vol. 20, no 3, p. 273-297.

[14] Cua, G., & Heaton, T. *The Virtual Seismologist (VS) method: A Bayesian approach to earthquake early warning.* En Earthquake early warning systems. Springer, Berlin, Heidelberg, 2007. p. 97-132.

[15] Cybenko, G. *Approximation by superpositions of a sigmoidal function.* Mathematics of control, signals and systems, 1989, vol. 2, no 4, p. 303-314.

[16] Egozcue, J. J., & Rüttener, E. *Bayesian techniques for seismic hazard assessment using imprecise data.* Natural hazards, 1996, vol. 14, no 2-3, p. 91-112.

[17] Fan, B., Lu, X., & Li, H. X. *Probabilistic inference-based least squares support vector machine for modeling under noisy environment.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016, vol. 46, no 12, p. 1703-1710.

[18] Farag, W., & Tawfik, A. *On fuzzy model identification and the gas furnace data.* In Proceedings of the IASTED International Conference Intelligent Systems and Control. 2000. p. 14-16.

[19] Fischer, A., & Igel, C. *Training restricted Boltzmann machines: An introduction.* Pattern Recognition, 2014, vol. 47, no 1, p. 25-39.

[20] Frigola, R., & Rasmussen, C. E. *Integrated pre-processing for Bayesian nonlinear system identification with Gaussian processes.* En 52nd IEEE Conference on Decision and Control. IEEE, 2013. p. 5371-5376.

[21] Gal, Y., Islam, R., & Ghahramani, Z. *Deep bayesian active learning with image data.* En International Conference on Machine Learning. PMLR, 2017. p. 1183-1192.

[22] Gernoth, K. A., & Clark, J. W. *Neural networks that learn to predict probabilities: Global models of nuclear stability and decay.* Neural Networks, 1995, vol. 8, no 2, p. 291-311.

[23] Gordon, J., & Hernández-Lobato, J. M. *Bayesian semisupervised learning with deep generative models.* arXiv preprint arXiv:1706.09751, 2017.

[24] Griffiths, T., & Alan Yuille. *A primer on probabilistic inference.* The probabilistic mind: Prospects for Bayesian cognitive science, 2008, p. 33-57.

[25] Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd Edition. Springer, 2009.

[26] Hill, T., O'Connor, M., & Remus, W. *Neural network models for time series forecasts.* Management science, 1996, vol. 42, no 7, p. 1082-1092.

[27] Hinton, G. E., Osindero, S., & Teh, Y. W. *A fast learning algorithm for deep belief nets.* Neural computation, 2006, vol. 18, no 7, p. 1527-1554.

[28] Hochreiter, S., & Schmidhuber, J. *Long short-term memory.* Neural computation, 1997, vol. 9, no 8, p. 1735-1780.

[29] Hoff, Peter D. *A first course in Bayesian statistical methods.* New York: Springer, 2009.

[30] Horn, R. A., & Johnson, C. R. *Matrix analysis.* Cambridge university press, 2012.

[31] Hornik, K., Stinchcombe, M., & White, H. *Multilayer feedforward networks are universal approximators.* Neural networks, 1989, vol. 2, no 5, p. 359-366.

[32] Huang, G. B., Zhu, Q. Y., & Siew, C. K. *Extreme learning machine: theory and applications.* Neurocomputing, 2006, vol. 70, no 1-3, p. 489-501.

[33] Jean, N., Xie, M., & Ermon, S. *Semi-supervised deep kernel learning.* In NIPS Bayesian deep learning workshop. 2016. p. 19-55.

[34] Kang, S. Y. *An investigation of the use of feedforward neural networks for forecasting.* 1991. Tesis Doctoral. Kent State University.

[35] Kendall, A., & Cipolla, R. *Geometric loss functions for camera pose regression with deep learning.* In Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 5974-5983.

[36] Khalil, H., & Grizzle, J. W. *Nonlinear Systems.* Prentice Hall, Upper Saddle River. 2002.

[37] Kolarik, T., & Rudorfer, G. *Time series forecasting using neural networks.* ACM Sigapl Apl Quote Quad, 1994, vol. 25, no 1, p. 86-94.

[38] Lambert, J., Sener, O., & Savarese, S. *Deep learning under privileged information using heteroscedastic dropout.* En Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. p. 8886-8895.

[39] Längkvist, M., Karlsson, L., & Loutfi, A. *A review of unsupervised feature learning and deep learning for time-series modeling.* Pattern Recognition Letters, 2014, vol. 42, p. 11-24.

[40] LeCun, Y., Bengio, Y., & Hinton, G. *Deep learning*. nature, 2015, vol. 521, no 7553, p. 436-444.

[41] LeCun, Y., et al. *Backpropagation applied to handwritten zip code recognition*. Neural computation, 1989, vol. 1, no 4, p. 541-551.

[42] Lee, J., et al. *Deep neural networks as gaussian processes*. arXiv preprint arXiv:1711.00165, 2017.

[43] Liang, F. *Bayesian neural networks for nonlinear time series forecasting*. Statistics and computing, 2005, vol. 15, no 1, p. 13-29.

[44] Liang, F., Liu, C., & Carroll, R. *Advanced Markov chain Monte Carlo methods: learning from past samples*. John Wiley & Sons, 2011.

[45] Ljung L., *System identification-theory for the user*, 2nd edition ptr prentice-hall. Upper Saddle River, NJ, 1999.

[46] Lu, Y., Huang, B., & Khatibisepehr, S. *A variational Bayesian approach to robust identification of switched ARX models*. IEEE transactions on cybernetics, 2015, vol. 46, no 12, p. 3195-3208.

[47] MacKay, D. J. *Bayesian methods for neural networks: Theory and applications*. 1995.

[48] Meier, M. A., Heaton, T., & Clinton, J. *The Gutenberg algorithm: Evolutionary Bayesian magnitude estimates for earthquake early warning with a filter bank*. Bulletin of the Seismological Society of America, 2015, vol. 105, no 5, p. 2774-2786.

[49] Müller, P., & Insua, D. R. *Issues in Bayesian analysis of neural network models*. Neural Computation, 1998, vol. 10, no 3, p. 749-770.

[50] Moreno-Noguer, F. *Multiple cue integration for robust tracking in dynamic environments: application to video relighting*. Universitat Politècnica de Catalunya. 2005.

[51] Murphy, K. *A brief introduction to graphical models and bayesian networks*, Rap. tech, 2001, vol. 96, p. 1-19.

[52] Neal, R. M. *Bayesian learning for neural networks*. Springer Science & Business Media, 2012.

[53] Osband, I. *Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout*. En NIPS workshop on bayesian deep learning. 2016.

[54] Park, Y. R., Murray, T. J., & Chen, C. *Predicting sun spots using a layered perceptron neural network*. IEEE Transactions on Neural Networks, 1996, vol. 7, no 2, p. 501-505.

[55] Peterka V., *Bayesian approach to system identification*. En Trends and Progress in System identification. Pergamon, 1981. p. 239-304.

[56] Qiu, Xueheng, et al. *Ensemble deep learning for regression and time series forecasting*. En 2014 IEEE symposium on computational intelligence in ensemble learning (CIEL). IEEE, 2014. p. 1-6.

[57] Rao, T. S., & Gabr, M. M. *An introduction to bispectral analysis and bilinear time series models*. Springer Science & Business Media, 2012.

[58] Rohekar, R. Y. Y., Koren, G., Nisimov, S., & Novik, G. *Unsupervised Deep Structure Learning by Recursive Independence Testing*.

[59] Roth, W., & Pernkopf, F. *Variational inference in neural networks using an approximate closed-form objective*. In Proceedings of the NIPS* 2016 Workshop on Bayesian Deep Learning. 2016.

[60] Russell, S., Norvig, P., & Intelligence, A. *Knowledge and reasoning: A modern approach*. En Artificial Intelligence. Prentice-Hall, 1995. p. 27.

[61] Sainath, T. N., et al. *Low-rank matrix factorization for deep neural network training with high-dimensional output targets*. In 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013. p. 6655-6659.

[62] Scherer, D., Müller, A., & Behnke, S. *Evaluation of pooling operations in convolutional architectures for object recognition*. In International conference on artificial neural networks. Springer, Berlin, Heidelberg, 2010. p. 92-101.

[63] Schoukens, Maarten, et al. *Cascaded tanks benchmark combining soft and hard nonlinearities*. En Workshop on nonlinear system identification benchmarks. 2016. p. 20-23.

[64] Singh, S., Gupta, A., & Efros, A. A. *Unsupervised discovery of mid-level discriminative patches*. In European Conference on Computer Vision (pp. 73-86). Springer, Berlin, Heidelberg. 2012.

[65] Snoek, J., et al. *Scalable bayesian optimization using deep neural networks.* En International conference on machine learning. PMLR, 2015. p. 2171-2180.

[66] Stavrakakis, G. N., & Tselentis, G. A. *Bayesian probabilistic prediction of strong earthquakes in the main seismogenic zones of Greece.* Boll. Geofis. Teor. Applic, 1987, vol. 113, p. 51-63.

[67] Stavrakakis, G. N., & Drakopoulos, J. *Bayesian probabilities of earthquake occurrences in Greece and surrounding areas.* pure and applied geophysics, 1995, vol. 144, no 2, p. 307-319.

[68] Su, C. T., & Chou, C. J. *A neural network-based approach for statistical probability distribution recognition.* Quality Engineering, 2006, vol. 18, no 3, p. 293-297.

[69] Sundermeyer, M., Schlüter, R., & Ney, H. *LSTM neural networks for language modeling.* En Thirteenth annual conference of the international speech communication association. 2012.

[70] Srivastava, N., et al. *Dropout: a simple way to prevent neural networks from overfitting.* The journal of machine learning research, 2014, vol. 15, no 1, p. 1929-1958.

[71] Telesca, L., Fat-Elbary, R., Stabile, T. A., Haggag, M., & Elgabry, M. *Dynamical characterization of the 1982–2015 seismicity of Aswan region (Egypt).* Tectonophysics, 2017, vol. 712, p. 132-144.

[72] Telesca, L., and. Lapenna, V., *Measuring multifractality in seismic sequences.* Tectonophysics, 2006, vol. 423, no 1-4, p. 115-123.

[73] Telesca, L.,et al. Multiparametric statistical investigation of seismicity occurred at El Hierro (Canary Islands) from 2011 to 2014. Tectonophysics, 2016, vol. 672, p. 121-128.

[74] Tong, H. *Non-linear time series: a dynamical system approach.* Oxford University Press, 1990.

[75] Tong, H., & Lim, K. S. *Threshold autoregression, limit cycles and cyclical data.* En Exploration Of A Nonlinear World: An Appreciation of Howell Tong's Contributions to Statistics. 2009. p. 9-56.

[76] Torres, M. D. S. *La inversa de Penrose.* In Anales de estudios económicos y empresariales (No. 1, pp. 299-308). Servicio de Publicaciones. 1986.

[77] Van der Westhuizen, J., & Lasenby, J. *Combining sequential deep learning and variational Bayes for semi-supervised inference.*

[78] van Stiphout, T., J. Zhuang, and D. Marsan, *Seismicity declustering*, Community Online Resource for Statistical Seismicity Analysis. 2012 doi:10.5078/corssa- 52382934. Available at http://www.corssa.org.

[79] Wang, H., Wang, N., & Yeung, D. Y. *Collaborative deep learning for recommender systems.* In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015. p. 1235-1244.

[80] Wang, H., Shi, X., & Yeung, D. Y. *Relational stacked denoising autoencoder for tag recommendation.* En Twenty-ninth AAAI conference on artificial intelligence. 2015.

[81] Wang, H., & Yeung, D. Y. *Towards Bayesian deep learning: A framework and some existing methods.* IEEE Transactions on Knowledge and Data Engineering, 2016, vol. 28, no 12, p. 3395-3408.

[82] Wang, J. P., et al. *Bayesian analysis on earthquake magnitude related to an active fault in Taiwan.* Soil Dynamics and Earthquake Engineering, 2015, vol. 75, p. 18-26.

[83] Weigend, A. S., Huberman, B. A., & Rumelhart, D. E. *Predicting the future: A connectionist approach.* International journal of neural systems, 1990, vol. 1, no 03, p. 193-209.

[84] Wilson, A. G., Knowles, D. A., & Ghahramani, Z. Gaussian process regression networks. arXiv preprint arXiv:1110.4411. 2011.

[85] Yin, L., Andrews, J., & Heaton, T. *Rapid earthquake discrimination for earthquake early warning: A Bayesian probabilistic approach using three-component single-station waveforms and seismicity forecast.* Bulletin of the Seismological Society of America, 2018, vol. 108, no 4, p. 2054-2067.

[86] Yu, W., & de la Rosa, E. Neural Modeling With Guaranteed Input-Output Probability Distributions. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2020.

[87] Zeiler, M. D., & Fergus, R. *Visualizing and understanding convolutional networks.* En European conference on computer vision. Springer, Cham, 2014. p. 818-833.

[88] Zhang, C., Tang, J., Li, H., Yang, C., Zhu, S., & Jin, R. *An Asynchronous Variance Reduced Framework for Efficient Bayesian Deep Learning.*