

**UNIVERSIDAD DE CUENCA**



**FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA**

**“MODELO PREDICTIVO DE FALLAS EN ALIMENTADORES PRIMARIOS DE  
CONCESIÓN DE LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR  
USANDO APRENDIZAJE PROFUNDO DE MÁQUINA”**

**Trabajo de titulación previo a  
obtención del título de Ingeniero  
Eléctrico**

**AUTORES:**

Juan Fernando Segarra López

C.I: 0105535413

Pedro Sebastian Andrade Veloz

C.I: 0502445505

**DIRECTOR:**

Dr. Ing. Santiago Patricio Torres Contreras

C.I: 0102448958

**CUENCA- ECUADOR**

**2017**



## RESUMEN

Este trabajo de titulación propone la predicción de interrupciones no programadas en alimentadores primarios de concesión de la Empresa Regional Centro Sur C.A. (CENTROSUR) usando Aprendizaje Profundo de Máquina. El modelo, basado en redes neuronales, ha sido poco explorado en la industria de la distribución de energía eléctrica, tanto en el ámbito local como internacional. Para la elaboración del modelo, se recopilaron y analizaron datos históricos de la empresa de los últimos cinco años con información de las interrupciones no programadas. La validación del modelo computacional de predicción se realizó usando información del alimentador #521 de la Subestación (S/E) 5 de la CENTROSUR, que es donde se evidencia la mayor incidencia de fallas.

**Palabras claves:** Aprendizaje profundo de máquina, Energía Eléctrica, Distribución eléctrica, Interrupciones no programadas (fallas), Mantenimiento predictivo.



## ABSTRACT

This work proposes the forecasting of failures in primary feeders of Empresa Eléctrica Regional Centro Sur C.A. (CENTROSUR) using Deep Learning. The model, based on neural networks, has not been wide explored in the electricity distribution industry, both locally and internationally. For the development of the model, the company's historical data from the last five years about failures was collected and analyzed. The validation of the forecasting model was performed using data from the #521 primary feeder of the Substation (S/E) 5 of CENTROSUR, where the highest incidence of faults is evidenced.

**Key words:** Deep Learning, Electric Power, Electrical distribution, Failures, Predictive maintenance.



## ÍNDICE DE CONTENIDO

|  |           |
|--|-----------|
| <b>CAPÍTULO 1</b> .....                                      | <b>17</b> |
| <b>1 INTRODUCCIÓN</b> .....                                  | <b>17</b> |
| 1.1 ANTECEDENTES.....  | 17        |
| 1.2 JUSTIFICACIÓN.....                                       | 18        |
| 1.3 ALCANCE .....  | 18        |
| 1.4 OBJETIVOS.....   | 19        |
| 1.4.1 OBJETIVO GENERAL .....                                 | 19        |
| 1.4.2 OBJETIVOS ESPECÍFICOS.....                             | 19        |
| 1.5 CONTENIDO .....  | 19        |
| <b>CAPÍTULO 2</b> .....                                      | <b>21</b> |
| <b>2 INTERRUPCIONES EN LOS SISTEMAS DE DISTRIBUCIÓN.....</b> | <b>21</b> |
| 2.1 SISTEMA DE DISTRIBUCIÓN ELÉCTRICA.....                   | 21        |
| 2.2 INTERRUPCIONES EN EL SISTEMA DE DISTRIBUCIÓN ELÉCTRICA   |           |
| 22   |           |
| 2.2.1 Interrupciones programadas .....                       | 22        |
| 2.2.2 Interrupciones no programadas .....                    | 23        |
| 2.2.2.1 Cortocircuitos.....                                  | 23        |
| 2.2.2.1.1 Falla trifásica .....                              | 24        |
| 2.2.2.1.2 Falla bifásica .....                               | 24        |
| 2.2.2.1.3 Falla bifásica con contacto a tierra .....         | 25        |
| 2.2.2.1.4 Falla monofásica .....                             | 25        |
| 2.2.2.2 Sobretensión.....                                    | 26        |
| 2.2.2.3 Transitorios .....                                   | 26        |
| 2.2.2.3.1 Fenómenos Oscilatorios .....                       | 26        |
| 2.2.2.3.2 Impulsos.....                                      | 26        |
| 2.2.2.4 Pérdida de aislamiento .....                         | 26        |
| 2.3 INFORMACIÓN DE FALLAS DE LA EMPRESA ELÉCTRICA            |           |
| REGIONAL CENTROSUR C.A.....                                  | 27        |
| 2.3.1 DATOS ESTADÍSTICOS DE LA EMPRESA.....                  | 27        |



|                         |  |           |
|-------------------------|--|-----------|
| 2.4                     | NORMAS DE CALIDAD EN EL SERVICIO ELÉCTRICO DE DISTRIBUCIÓN .....                                   | 27        |
| 2.5                     | DISPOSICIONES DE REGULACIONES VIGENTES .....   | 30        |
| <b>CAPÍTULO 3 .....</b> |  | <b>32</b> |
| <b>3</b>                | <b>APRENDIZAJE PROFUNDO DE MÁQUINA.....</b>  | <b>32</b> |
| 3.1                     | INTRODUCCIÓN.....  | 32        |
| 3.2                     | FUNDAMENTOS .....  | 34        |
| 3.3                     | ARQUITECTURAS.....   | 37        |
| 3.3.1                   | El perceptrón simple .....   | 38        |
| 3.3.2                   | El perceptrón multicapa .....  | 39        |
| 3.3.3                   | Redes neuronales recurrentes .....   | 41        |
| 3.3.4                   | Redes de memoria de largo plazo.....   | 43        |
| 3.4                     | APRENDIZAJE AUTOMÁTICO.....  | 45        |
| 3.4.1                   | Aprendizaje de máquina .....   | 45        |
| 3.4.2                   | Aprendizaje profundo de máquina (APM) .....  | 46        |
| 3.5                     | HERRAMIENTAS DE SOFTWARE USADAS PARA LA REDES NEURONALES APLICADAS A LA PREDICCIÓN DE FALLAS ..... | 47        |
| <b>CAPÍTULO 4 .....</b> |  | <b>49</b> |
| <b>4</b>                | <b>APLICACIÓN DEL MODELO.....</b>  | <b>49</b> |
| 4.1                     | INTRODUCCIÓN.....  | 49        |
| 4.2                     | DESCRIPCIÓN DEL MODELO .....   | 51        |
| 4.3                     | PROTOTIPO DEL MODELO PROPUESTO .....   | 52        |
| 4.3.1                   | Modelo para una característica.....  | 52        |
| 4.4                     | MODELO PARA LA PREDICCIÓN DE FALLAS EN UN ALIMENTADOR .....  | 56        |
| 4.4.1                   | Modelo para varias características .....   | 56        |
| 4.4.1.1                 | Modelo sin capas ocultas .....   | 59        |
| 4.4.1.1.1               | Variación del hiper parámetro batch sin capas ocultas.....   | 59        |
| 4.4.1.2                 | Modelo de una capa oculta: .....   | 62        |
| 4.4.1.2.1               | Variación del hiper parámetro batch con una capa oculta ..   | 62        |
| 4.4.1.3                 | Modelo de dos capas ocultas .....  | 64        |
| 4.4.1.3.1               | Variación del hiper parámetro batch con dos capas ocultas .....                                    | 64        |



|   |  |           |
|---|--|-----------|
| 4.4.1.4   | El error versus el aprendizaje por lotes.....                                      | 66        |
| 4.4.1.5   | Funciones de activación en la capa de salida.....                                  | 68        |
| 4.5   | PREDICCIONES NULAS .....   | 69        |
| 4.5.1.1   | Prueba con parámetros aleatorios .....   | 69        |
| 4.6   | RESULTADOS EN ORDENADORES CON DIFERENTE PLATAFORMA DE<br>HARDWARE Y SOFTWARE ..... | 70        |
| <b>CAPÍTULO 5</b>   | .....  | <b>72</b> |
| <b>5</b>  | <b>CONCLUSIONES Y RECOMENDACIONES</b> .....  | <b>72</b> |
| <b>BIBLIOGRAFÍA</b>   | .....  | <b>75</b> |
| <b>ANEXOS</b>   | .....  | <b>80</b> |
| ANEXO A   | .....  | 80        |
| TABLAS QUE REGISTRAN EL NÚMERO INTERRUPCIONES NO<br>PROGRAMADAS QUE SE HAN GENERADO EN LOS AÑOS 2011, 2012,<br>2013, 2014 Y 2015..... |  | 80        |
| ANEXO B   | .....  | 85        |
| DATOS ESTADÍSTICOS DE LA EMPRESA ELÉCTRICA REGIONAL CENTRO<br>SUR C.A.....  |  | 85        |
| ANEXO C   | .....  | 86        |
| CODIGO DEL MODELO COMPUTACIONAL DE PREDICCIÓN DE FALLAS..   |  | 86        |



## INDICE DE FIGURAS

|   |    |
|---|----|
| <b>Figura 2.1</b> Configuración típica de un Sistema Eléctrico de Potencia .....                  | 21 |
| <b>Figura 2.2</b> Falla trifásica .....   | 24 |
| <b>Figura 2.3</b> Falla bifásica .....  | 24 |
| <b>Figura 2.4</b> Falla bifásica a tierra .....   | 25 |
| <b>Figura 2.5</b> Falla monofásica .....  | 25 |
| <b>Figura 3.1</b> Ilustración de una neurona biológica .....                                      | 33 |
| <b>Figura 3.2</b> Red de múltiples entradas cuyos nodos son neuronas artificiales .....           | 33 |
| <b>Figura 3.3</b> Red de múltiples salidas cuyos nodos son neuronas artificiales .....            | 33 |
| <b>Figura 3.4</b> Secciones de una red neuronal .....   | 34 |
| <b>Figura 3.5</b> Ilustración de una neurona artificial .....                                     | 35 |
| <b>Figura 3.6</b> Gráfico cartesiano de la función bipolar .....                                  | 36 |
| <b>Figura 3.7</b> Gráfico cartesiano de la función tanh .....                                     | 36 |
| <b>Figura 3.8</b> Gráfico cartesiano de la función $\sigma$ .....                                 | 37 |
| <b>Figura 3.9</b> Gráfico cartesiano de la función relu .....                                     | 37 |
| <b>Figura 3.10</b> Región correspondiente a la decisión de un perceptrón simple .....             | 38 |
| <b>Figura 3.11</b> Ajuste y evolución de las decisiones de un perceptrón simple .....             | 39 |
| <b>Figura 3.12</b> Región de decisión de una RN con arquitectura PMC .....                        | 41 |
| <b>Figura 3.13</b> Una RNN y el despliegue de su cálculo en el tiempo .....                       | 42 |
| <b>Figura 3.14</b> Esquema general del modelo computacional propuesto .....                       | 48 |
| <b>Figura 4.1</b> Incremento poblacional de Ecuador, Perú y Bolivia .....                         | 49 |
| <b>Figura 4.2</b> Datos reales de fallas en un alimentador .....                                  | 50 |
| <b>Figura 4.3</b> Aproximación mediante series de tiempo de las fallas en un alimentador .....    | 50 |
| <b>Figura 4.4</b> Diagrama de flujo del modelo propuesto .....                                    | 52 |
| <b>Figura 4.5</b> Predicción de una onda senoidal, epochs = 2 .....                               | 53 |
| <b>Figura 4.6</b> Predicción de una onda senoidal, epochs = 4 .....                               | 54 |
| <b>Figura 4.7</b> Predicción de una onda senoidal, epochs = 8 .....                               | 54 |
| <b>Figura 4.8</b> Predicción de una onda senoidal, epochs = 16 .....                              | 55 |
| <b>Figura 4.9</b> Pérdidas vs las iteraciones en el modelo de APM .....                           | 55 |
| <b>Figura 4.10</b> Pérdidas versus iteraciones en el modelo de varias características .....       | 58 |
| <b>Figura 4.11</b> Predicciones de coordenadas $X$ e $Y$ , red sin capas ocultas, batch = 2 ..... | 60 |
| <b>Figura 4.12</b> Predicciones de coordenadas $X$ e $Y$ , red sin capas ocultas, batch = 5 ..... | 60 |

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz



**Figura 4.13** Predicciones de coordenadas  $X$  e  $Y$ , red sin capas ocultas, batch = 10 ..... 60

**Figura 4.14** Predicciones de coordenadas  $X$  e  $Y$ , red sin capas ocultas, batch = 15 ..... 61

**Figura 4.15** Predicciones de coordenadas  $X$  e  $Y$ , red de una capa oculta, batch = 2 ..... 62

**Figura 4.16** Predicciones de coordenadas  $X$  e  $Y$ , red de una capa oculta, batch = 5 ..... 62

**Figura 4.17** Predicciones de coordenadas  $X$  e  $Y$ , red de una capa oculta, batch = 10 ..... 63

**Figura 4.18** Predicciones de coordenadas  $X$  e  $Y$ , red de una capa oculta, batch = 15 ..... 63

**Figura 4.19** Predicciones de coordenadas  $X$  e  $Y$ , red de dos capas ocultas, batch = 2 ..... 64

**Figura 4.20** Predicciones de coordenadas  $X$  e  $Y$ , red de dos capas ocultas, batch = 5 ..... 65

**Figura 4.21** Predicciones de coordenadas  $X$  e  $Y$ , red de dos capas ocultas, batch = 10 ..... 65

**Figura 4.22** Predicciones de coordenadas  $X$  e  $Y$ , red de dos capas ocultas, batch = 15 ..... 65

**Figura 4.23** RMSE vs batch en el entrenamiento de dos capas ocultas para  $X$  ..... 66

**Figura 4.24** RMSE vs batch en las pruebas de dos capas ocultas para  $X$  ..... 67

**Figura 4.25** RMSE vs batch en el entrenamiento de dos capas ocultas para  $Y$  ..... 67

**Figura 4.26** RMSE vs batch en las pruenas de dos capas ocultas para  $Y$  ..... 68

**Figura 4.27** Predicción de coordenadas  $X$  e  $Y$  con parámetros aleatorios ..... 70

**Figura 4.28** Predicción de la coordenada  $Y$  (izquierda: GPU, derecha: CPU) ..... 71



## INDICE DE TABLAS

|   |    |
|---|----|
| <b>Tabla 4.1</b> Error de una red sin capas ocultas para diferentes batch, predicción de coordenadas $X$ .....  | 59 |
| <b>Tabla 4.2</b> Error de una red sin capas ocultas para diferentes batch, predicción de coordenadas $Y$ .....  | 59 |
| <b>Tabla 4.3</b> Error de una red con una capa oculta para diferentes batch, predicción de coordenadas $X$ .....  | 61 |
| <b>Tabla 4.4</b> Error de una red con una capa oculta para diferentes batch, predicción de coordenadas $Y$ .....  | 62 |
| <b>Tabla 4.5</b> Error de una red con dos capas ocultas para diferentes batch, predicción de coordenadas $X$ .....  | 64 |
| <b>Tabla 4.6</b> Error de una red con dos capas ocultas para diferentes batch, predicción de coordenadas $Y$ .....  | 64 |
| <b>Tabla 4.7</b> Resultados del error en base al número de capas ocultas de la red y el batch para la predicción de coordenadas $X$ .....                       | 66 |
| <b>Tabla 4.8</b> Resultados del error en base al número de capas ocultas de la red y el batch para la predicción de coordenadas $Y$ .....                       | 67 |
| <b>Tabla 4.9</b> Error para diferentes funciones de activación en la capa de salida de un modelo de dos capas ocultas de la predicción de coordenada $X$ .....  | 68 |
| <b>Tabla 4.10</b> Error para diferentes funciones de activación en la capa de salida de un modelo de dos capas ocultas de la predicción de coordenada $Y$ ..... | 68 |
| <b>Tabla 4.11</b> Error para una red configurada sin criterio de selección de parámetros .....  | 69 |
| <b>Tabla 4.12</b> Comparación de los diferentes ordenadores .....   | 70 |
| <b>Tabla 4.13</b> Errores en la predicción de coordenadas $Y$ en 2 ordenadores .....  | 71 |
| <b>Tabla A.1</b> Total, de interrupciones registradas para el año 2011 .....  | 79 |
| <b>Tabla A.2</b> Total, de interrupciones registradas para el año 2012 .....  | 80 |
| <b>Tabla A.3</b> Total, de interrupciones registradas para el año 2013 .....  | 81 |
| <b>Tabla A.4</b> Total, de interrupciones registradas para el año 2014 .....  | 82 |
| <b>Tabla A.5</b> Total, de interrupciones registradas para el año 2015 .....  | 83 |
| <b>Tabla B.1</b> Datos estadísticos de la Empresa Eléctrica Regional Centro Sur C.A .....   | 85 |



**Cláusula de licencia y autorización para publicación en el Repositorio  
Institucional**

Juan Fernando Segarra López, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación **“MODELO PREDICTIVO DE FALLAS EN ALIMENTADORES PRIMARIOS DE CONCESIÓN DE LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR USANDO APRENDIZAJE PROFUNDO DE MÁQUINA”**, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, noviembre del 2017

Juan Fernando Segarra López

C.I: 0105535413



### Cláusula de Propiedad Intelectual

Juan Fernando Segarra López, autor de la tesis **“MODELO PREDICTIVO DE FALLAS EN ALIMENTADORES PRIMARIOS DE CONCESIÓN DE LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR USANDO APRENDIZAJE PROFUNDO DE MÁQUINA”**, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad del autor.

Cuenca, octubre del 2017

---

Juan Fernando Segarra López

C.I: 0105535413



**Cláusula de licencia y autorización para publicación en el Repositorio  
Institucional**

Pedro Sebastian Andrade Veloz, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "**MODELO PREDICTIVO DE FALLAS EN ALIMENTADORES PRIMARIOS DE CONCESIÓN DE LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR USANDO APRENDIZAJE PROFUNDO DE MÁQUINA**", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, noviembre del 2017

Pedro Sebastian Andrade Veloz

C.I: 050244550-5



### Cláusula de Propiedad Intelectual

Pedro Sebastian Andrade Veloz, autor de la tesis **“MODELO PREDICTIVO DE FALLAS EN ALIMENTADORES PRIMARIOS DE CONCESIÓN DE LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR USANDO APRENDIZAJE PROFUNDO DE MÁQUINA”**, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad del autor.

Cuenca, octubre del 2017

Pedro Sebastian Andrade Veloz

C.I: 050244550-5



## **AGRADECIMIENTOS**

En el presente proyecto de titulación agradecemos a Dios por brindarnos un día más de vida y llenarnos de bendiciones. A nuestros padres por el sacrificio que día a día han realizado por nuestra formación profesional y personal. A los docentes de la Universidad de Cuenca por compartir con nosotros su experiencia y su conocimiento. A Santiago Torres. PhD, Director de tesis, quien por su compromiso y apoyo brindado para culminar este trabajo ha servido de inspiración para incursionar en nuevos campos investigativos. A nuestros compañeros de aula por las alegrías vividas. A nuestra alma máter, la Universidad de Cuenca por brindarnos la oportunidad de haber pertenecido a ella.

Los Autores



## **AGRADECIMIENTO**

Agradezco a mis padres Darwin e Isabel, quien son las personas más importantes en mi vida, gracias por su apoyo he logrado culminar mi educación en esta etapa. A mi hermano Josué quien impulsa en mí el deseo de lograr la excelencia académica.

## **DEDICATORIA**

En memoria de mis abuelos Humberto, Rosa y Víctor,  
Juan Fernando



## **AGRADECIMIENTO**

Agradezco de todo corazón a mi padre Pedro, quien con su entusiasmo ha sido el motor principal para que luche por mis propósitos. A mis hermanos Oscar y Daniel por todos los consejos y protección a lo largo de estos años. A mi familia que me supo acoger en su debido momento. A todas esas personas que han formado parte de mi vida y han aportado un granito de arena para que esto sea posible.

## **DEDICATORIA**

En memoria de Susana Veloz y Margoth Veloz,  
Pedro



## CAPÍTULO 1

### 1 INTRODUCCIÓN

#### 1.1. ANTECEDENTES

Las empresas distribuidoras de energía eléctrica en general, y en particular la Empresa Eléctrica Regional Centro Sur C.A. (CENTROSUR), presentan un número importante de interrupciones no programadas (fallas) en cada uno de sus alimentadores primarios todos los meses del año. Son varias las causas de estas fallas, no obstante, algunas de estas interrupciones se repiten en una fecha y lugar donde previamente se han originado. Un ejemplo de esto es que, según funcionarios de la CENTROSUR, existen fallas que suceden de forma recurrente por causas de la vegetación en ciertas zonas rurales. En [1] se evidencia que las fallas debido a la vegetación son un problema común en varios sistemas de distribución. Una cantidad importante de estas interrupciones presentan patrones repetitivos.

En los últimos años, se han desarrollado herramientas de aprendizaje profundo de máquina (APM) muy efectivas, que son una forma especial de algoritmos de aprendizaje de máquina (AAM) optimizadas para resolver ciertos problemas como predicciones de series de tiempo, procesamiento de imágenes, análisis de texto, reconocimiento de voz, etc. Dado que actualmente la mayoría de equipos computacionales cuentan con un alto poder de cómputo, el APM es capaz de aprovechar la capacidad de procesamiento computacional en el hardware moderno sin comprometer al usuario con recursos caros y difíciles de conseguir. Adicionalmente, esta herramienta se ha usado para resolver problemas relacionados a sistemas de energía eléctrica. En [2] se hace uso de la herramienta para predicciones eólicas en base a datos meteorológicos. De igual manera en [3] se usó la herramienta para predecir la demanda de energía eléctrica en México. En [4] se usó aprendizaje de máquina para realizar predicciones de mantenimiento tomando en cuenta factores estacionales e incremento de la carga.



## 1.2. JUSTIFICACIÓN

El sistema de distribución (SD) de energía eléctrica, parte fundamental dentro de un sistema eléctrico de potencia (SEP), suministra electricidad a los diferentes tipos de usuarios. Las fallas del SD tales como salidas de servicio de alimentadores, daños en transformadores, fallos en la red, entre otras, son eventos no deseados y pueden causar la suspensión del servicio de energía en ciertas zonas urbanas y/o rurales que por defecto deben contar con energía eléctrica todo el tiempo. Estas interrupciones no programadas pueden ser causadas por cambios climáticos, sabotajes, accidentes de tránsito, calamidades en fauna o flora, descargas atmosféricas, etc. Además, los diferentes tipos de cargas conectadas en el sistema de distribución se pueden ver afectadas por la falta en la continuidad del servicio, y la interrupción de energía eléctrica puede acarrear graves pérdidas económicas tanto para la empresa de suministro como para el cliente.

Debido a la importancia de tratar con estas interrupciones no programadas, es deseable para la empresa distribuidora contar con herramientas que permitan ubicar la falla en tiempo y lugar de forma anticipada para poder llevar a cabo acciones específicas de mantenimiento preventivo. El problema no es nuevo en el sector eléctrico, sin embargo, las soluciones para el mismo son poco exploradas hasta la actualidad. La mayoría de investigaciones en este campo no llevan más de 20 años. En [5], uno de los escasos trabajos encontrados en el tema, se utilizó AAM para predicción de fallas en alimentadores de la empresa Consolidated Edison (CoEdison) en Manhattan-New York.

## 1.3. ALCANCE

En este trabajo de titulación se hará uso de técnicas de APM para la predicción de fallas, utilizando como entradas la información histórica de interrupciones no programadas de la CENTROSUR. Mediante el diseño de una red neuronal se obtendrá, la ubicación y fecha aproximadas de posibles fallas en alimentadores primarios de la distribuidora. El presente proyecto asume que la ocurrencia de fallas se puede modelar de forma aproximada como una serie de



tiempo. El modelo computacional que se propone deberá ser capaz de aprender las características que determinan el comportamiento temporal de la ocurrencia de las fallas, para obtener como resultado la ubicación en espacio y tiempo de una posible falla representada por una curva de dos dimensiones.

## 1.4. OBJETIVOS

### 1.4.1. OBJETIVO GENERAL

Desarrollar un modelo computacional para predecir interrupciones no programadas, en tiempo y lugar, en áreas de concesión de la Empresa Eléctrica Regional Centro Sur C.A.

### 1.4.2. OBJETIVOS ESPECÍFICOS

- i. Desarrollar un modelo computacional aplicable que realice predicciones de fallas, arrojando como resultado la localización y fecha aproximada
- ii. Utilizar la herramienta de aprendizaje profundo de máquina para crear un modelo computacional capaz de realizar predicciones de la ocurrencia de fallas, usando los datos de un alimentador real de la empresa
- iii. Aportar a la empresa un método computacional útil para la programación de actividades de mantenimiento preventivo

## 1.5. CONTENIDO

Este trabajo dispone de cinco capítulos organizados de la siguiente manera:

En el **Capítulo 1** se introduce el problema de fallas en alimentadores de la CENTROSUR; se propone una posible solución al problema de predicción de fallas y se plantean los objetivos del trabajo de titulación.



En el **Capítulo 2** se estudia el sistema de distribución de la CENTROSUR, haciendo énfasis en los conceptos de interrupciones programadas y no programadas, y el historial que la empresa distribuidora presenta con respecto a la periodicidad de fallas.

Por otro lado, el **Capítulo 3** desarrolla conceptos matemáticos para la comprensión del algoritmo que se propone como solución al problema planteado.

El **Capítulo 4** presenta el algoritmo propuesto, al mismo tiempo que analiza su eficacia mediante comparaciones objetivas de varias predicciones y análisis de los resultados.

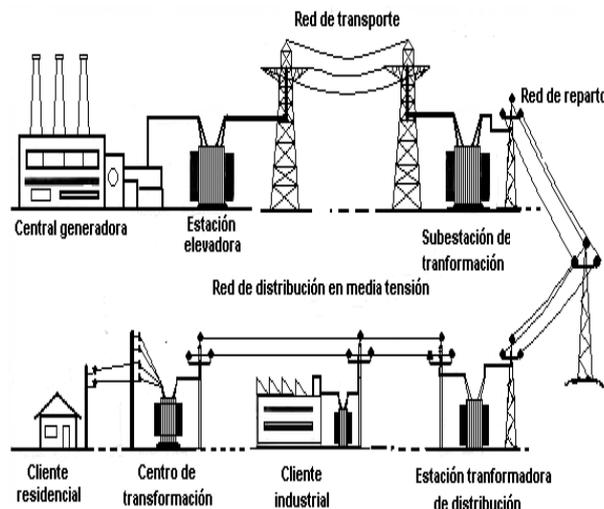
Por último, el **Capítulo 5** detalla las conclusiones y se expondrán recomendaciones de trabajo futuro en torno a lo estudiado a lo largo de este documento.

## CAPÍTULO 2

### 2 INTERRUPTIONES EN LOS SISTEMAS DE DISTRIBUCIÓN

#### 2.1 SISTEMA DE DISTRIBUCIÓN ELÉCTRICA

El Sistema Eléctrico de Potencia (SEP) debe ser capaz de generar, transmitir y distribuir energía eléctrica de forma segura y confiable. Un SEP se encuentra conformado por varios elementos como generadores, líneas de transmisión, subestaciones, redes de bajo y alto voltaje, etc.



**Figura 2.1** Configuración típica de un Sistema Eléctrico de Potencia [6]

En la Figura 2.1, se observa la estructura de un SEP dividida en tres etapas: generación, transmisión y distribución:

- i. **Generación:** Es la producción de energía eléctrica a través de fuentes de generación renovables y/o no renovables
- ii. **Transmisión:** Se encarga de transportar la energía producida a los centros de consumo



- iii. **Distribución:** Está conformada por alimentadores primarios y secundarios que transportan energía de medio y bajo voltaje

El sistema de distribución eléctrica conecta a los usuarios finales con el Sistema de Transmisión (ST). Parámetros como la tensión, corriente y frecuencia sirven para conocer el estado de operación de un SD.

Una empresa distribuidora, que es la que gestiona el sistema de distribución, tiene la misión de suministrar de forma continua y confiable la energía eléctrica, manteniendo ciertos estándares que en el Ecuador se rigen por el Art 2 de La Ley Orgánica del Servicio Público de Energía Eléctrica.

## 2.2 INTERRUPCIONES EN EL SISTEMA DE DISTRIBUCIÓN ELÉCTRICA

La operación de un SD de forma confiable y segura es de suma importancia para un país [7]- [8]. En el caso de ocurrencia de una falla, las empresas distribuidoras optan por mitigar la recurrencia de estas al disminuir los potenciales peligrosos y aislar la parte vulnerable del sistema. En un SD, muchas de las interrupciones son inevitables, y es porque algunas son accidentales.

Una interrupción producida en el SD puede clasificarse según el origen como: programada y no programada.

### 2.2.1 Interrupciones programadas

Las interrupciones programadas son aquellas que sucedieron bajo la supervisión y planeamiento de la empresa distribuidora

#### Características

- Interrupción planeada para mejorar y/o expandir la red



- Planificada para alargar la vida útil de ciertos elementos conectados al sistema
- Programada con anterioridad y supervisión

### 2.2.2 Interrupciones no programadas

Son fallas que suceden en circunstancias ajenas al programa de actividades de la empresa distribuidora

#### Características

- Causan calamidades, pudiendo estas derivar en daños a los seres vivos
- Son el origen del malestar y reclamo de los usuarios
- Situación de riesgo para el sistema en general
- Aumento de los daños en equipos dentro de las instalaciones eléctricas
- Producen paradas de procesos industriales sin previo aviso

Toda interrupción no programada o falla es causada por una condición especial del circuito. Sin embargo, no todas esas condiciones causan fallas; no al menos si se activan las protecciones del sistema de distribución o el problema es de corta duración. Estas condiciones pueden clasificarse de la siguiente manera:

#### 2.2.2.1 Cortocircuitos

Definimos cortocircuito al contacto accidental o conexión anormal en algún punto del sistema con diferente potencial, dando como resultado aumentos excesivos de corriente. Para prevenir que esto se propague desde el punto de falla al resto del circuito, se realizan estudios de coordinación de protecciones, cálculos de cortocircuitos y flujos de potencia. Lo cual permitirá aislar la parte afectada del resto de la red.

Los tipos de cortocircuitos que presentan pérdidas totales o parciales de tensión son las siguientes:

### 2.2.2.1.1 Falla trifásica

Se conoce como falla trifásica al contacto directo entre las tres fases del circuito, con incrementos de potenciales sumamente peligrosos.

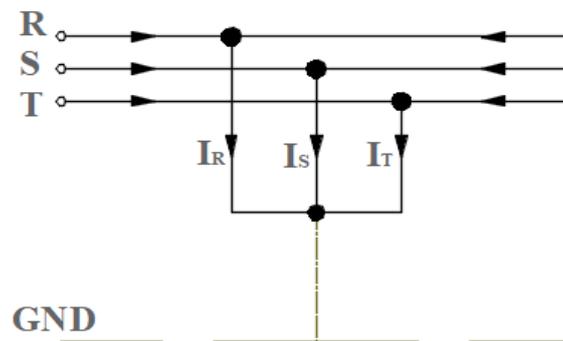


Figura 2.2 Falla trifásica

### 2.2.2.1.2 Falla bifásica

Se define a un cortocircuito bifásico como el contacto directo entre dos conductores de diferente fase o a la pérdida de aislamiento entre uno de ellos.

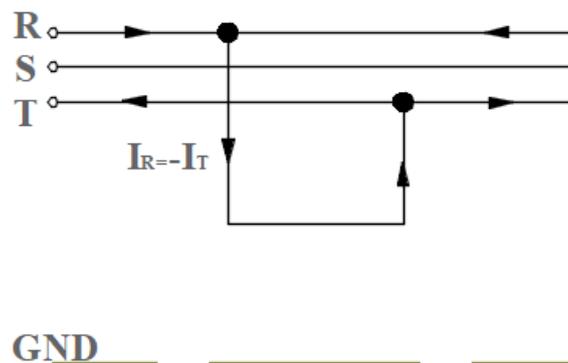


Figura 2.3 Falla bifásica

### 2.2.2.1.3 Falla bifásica con contacto a tierra

Un cortocircuito bifásico a tierra se comporta de la misma manera que una falla bifásica con la diferencia que, las pérdidas de energía son conducidas a tierra.

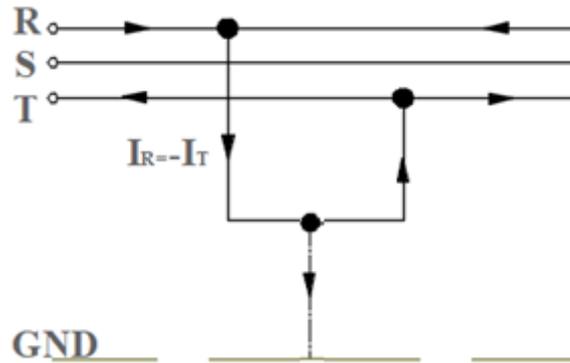


Figura 2.4 Falla bifásica a tierra

En la Figura 2.4 se puede observar como la corriente de falla conduce a tierra.

### 2.2.2.1.4 Falla monofásica

Un cortocircuito monofásico es producto de la conexión de una de las fases de un circuito con el conductor a tierra.

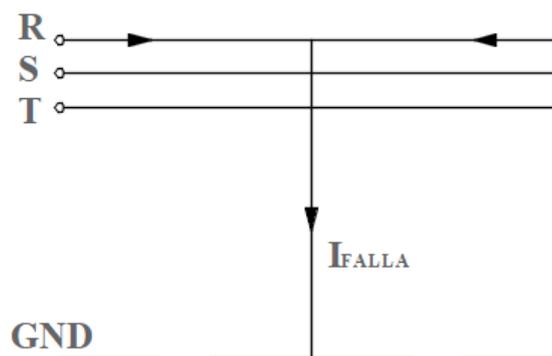


Figura 2.5 Falla monofásica



### 2.2.2.2 Sobretensión

La sobretensión es el incremento de voltaje de corta duración, medido entre dos conductores del circuito. El calentamiento del conductor y la pérdida de aislamiento es un indicador de sobretensión.

### 2.2.2.3 Transitorios

Los transitorios producen variaciones de corriente y voltaje; el problema puede presentarse de manera oscilatoria o por impulsos.

#### 2.2.2.3.1 Fenómenos Oscilatorios

Se reconoce el estado oscilatorio cuando el sistema registra una frecuencia de falla mayor a la frecuencia de operación [7]- [9].

En base a la frecuencia se distinguen:

- i. **Alta frecuencia** mayor a 500 kHz
- ii. **Media frecuencia** entre 5 a 500 kHz
- iii. **Baja frecuencia** menor a 5 kHz

#### 2.2.2.3.2 Impulsos

Involucran altas frecuencias y variaciones de corriente en dirección positiva y negativa [7].

### 2.2.2.4 Pérdida de aislamiento

La pérdida de aislamiento generalmente se origina por el envejecimiento de los elementos eléctricos conectados a la red.

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz



## **2.3 INFORMACIÓN DE FALLAS DE LA EMPRESA ELÉCTRICA REGIONAL CENTROSUR C.A.**

La CENTROSUR está ubicada en la provincia del Azuay, cantón Cuenca. Es la compañía encargada de distribuir energía eléctrica a gran parte del Azuay, Cañar y Morona Santiago con excepción de algunos cantones; abarca el 11.79% de territorio ecuatoriano [10]. Posee una extensión de 30.243  $km^2$  divididos en zonas (Zona 1, Zona 2, Zona 3) y, la Troncal, que se considera desde el año 2014. La mayor parte del área de concesión de la empresa ocupa la provincia de Morona Santiago con un margen del 62.84 % expresado de manera porcentual, seguido de Azuay, Cañar y Otros (Naranjal, El Guabo, Saraguro y el Piedrero) que ocupa el 0.78% [10].

### **2.3.1. Datos estadísticos de la empresa**

El Anexo A detalla datos históricos de interrupciones no programadas (clasificación que la empresa divide según el origen en red, climáticas y terceros). De acuerdo a esa información, el total de fallas se ha incrementado en un 9% para el 2016, siendo el año con mayor índice de interrupciones desde el 2008. Las fallas por cortocircuito se han incrementado a través de los años y son la principal razón de interrupciones no programadas de la red.

Las interrupciones por vegetación son de alta incidencia en las empresas distribuidoras de energía eléctrica, especialmente en la CENTROSUR, que muestra un índice de ocurrencia del 60% del total de las interrupciones no programadas.

## **2.4 NORMAS DE CALIDAD EN EL SERVICIO ELÉCTRICO DE DISTRIBUCIÓN**

La empresa se encarga de recopilar la información técnica y administrativa por medio del Departamento de Análisis y Sistemas de Distribución (SIGADE), información necesaria para medir los niveles de calidad de energía entregada en diferentes etapas de operación [11].



En cuanto al almacenamiento de información, la CENTROSUR dispone de una base de datos con interrupciones programadas y no programadas desde el año 2008, lo cual es exigencia de la Regulación No. CONELEC – 004/01. La información de cada falla se registra de la siguiente manera:

- Año: Para cada año se actualizará todas las interrupciones producidas en el sistema
- Mes de visualización: El sistema no estará libre de alguna perturbación, debido a esto, desde el mes de enero a diciembre de cada año el sistema se mantiene monitoreado diariamente, si el sistema SCADA detecta una interrupción el SIGADE registrará en el mes que fue causada
- Zona: Se registrará en una de las tres zonas donde se haya producido la falla del alimentador
- Alimentador: Se refiere al número del alimentador que fue afectado
- kVA instalado: Son los kVA instalados en el mes del análisis
- Nodo: Se registra el número del equipo de protección que se acciono en el momento de una falla
- Número de Interrupción: Se registra con un número de fallo actualizado diariamente
- Fecha de desconexión: Se registrará por año, mes, día y hora en el que origino la interrupción
- Potencia en kVA: Es la potencia en kVA que ha sido desconectada de la red
- Transformador: Se registra por el número del transformador



- Falla de Origen: Se registra de acuerdo a una interpretación generada por la EERCS, abarcando tres índices, en donde fue afectado el sistema (baja, media y alta tensión, etc.), tipo de daño (ramal, general y puntual.) y por su origen (interrupción programada o no programada y externa.)
- Tiempo en horas: Se registra el tiempo en horas que se demoró la desconexión
- Descripción de la causa: Se describe la causa que originó la interrupción (cortocircuito, accidente de tránsito, animales en la red, etc.)
- Causa: Se describe de manera más específica (red, climáticas, terceros.)
- Origen: Se registra si la interrupción fue programada (mantenimiento, etc.) o no programada (rayos, vegetación, etc.)
- Dirección: Se registra la avenida y calle donde se originó el punto de falla
- Coordenadas en X: Es la ubicación del equipo que estuvo más próximo a falla, expresado en longitud
- Coordenadas en Y: Es la ubicación del equipo que estuvo más próximo a la falla, expresado en latitud
- FMIK: Es la frecuencia media de interrupción (veces que los kVA promedio han sufrido por una interrupción.) referido al sistema
- FMIKZ: Es la frecuencia media de interrupción referido a la zona
- FMIKA: Es la frecuencia media de interrupción referido al alimentador



- TTMIK: Representa el tiempo total de interrupción (tiempo total en que los kVA promedio no tuvo servicio.) referido al sistema
- TTMIKZ: Es el tiempo total de interrupción referido a la zona
- TTMIKA: Es tiempo total de interrupción referido al alimentador
- Clientes: Número de consumidores afectados por cada falla
- Subsistema: Se registra a que nivel de voltaje (baja, media o alta tensión.) se genera la falla

## 2.5 DISPOSICIONES DE REGULACIONES VIGENTES

Las empresas eléctricas en el Ecuador, deberán cumplir con las normativas dictadas por el ARCONEL, la ley entró en vigencia el 16 de enero del 2016 para fortalecer el servicio estipulado por la Ley Orgánica del Servicio Público.

La Regulación No.004/01 de calidad de energía, dice que <<Se consideran todas las interrupciones mayores a 3 minutos, incluyendo las de origen externo, debidas a fallas en transmisión. No se consideran las interrupciones con duración igual o menor a 3 minutos>> [12]. La misma regulación también estipula lo siguiente: <<No se considera interrupciones de un consumidor en particular causadas en sus propias instalaciones, siempre y cuando no afecten a otros consumidores>> [12]. << Tampoco es considerada, pero si registrada toda interrupción debida a suspensiones generales del servicio, racionamientos, desconexiones de carga por baja frecuencia en el caso de las interrupciones generales del servicio sean producidos por la Empresa Distribuidora, estas serán registrados>> [12].

De acuerdo a lo anterior, las acciones que deberán cumplir las empresas distribuidoras según la Regulación No. CONELEC - 004/01 son [12]:

- i. El distribuidor debe llevar un registro histórico por lo menos los tres últimos años

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz



- ii. En los registros deberá estar especificado, fecha, ubicación, origen, hora de inicio y finalización de la interrupción
- iii. El distribuidor deberá entregar informes mensuales al ente regulador

El cumplimiento de estos tres puntos hizo posible la obtención de datos de interés para este trabajo.



## CAPÍTULO 3

### 3 APRENDIZAJE PROFUNDO DE MÁQUINA

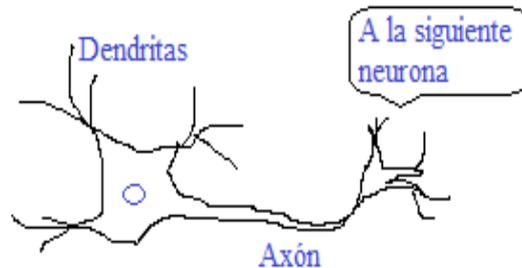
#### 3.1 INTRODUCCIÓN

El cerebro humano posee una compleja arquitectura de neuronas biológicas interconectadas entre sí, formando una extensa red donde la transmisión de impulsos nerviosos es posible mediante la sinapsis (región de comunicación entre los terminales de una neurona con otra). Esta red convierte reacciones de elementos electro-químicos en complejos conceptos para la mente como aprendizaje, memorización y razonamiento.

Un hombre en promedio cuenta con  $10^{11}$  neuronas y con  $10^4$  conexiones cada una. Sin embargo, el número de neuronas no es factor definitivo si se trata de medir la inteligencia. Las ballenas y los elefantes poseen casi el doble de cantidad de neuronas. En cuanto a la totalidad de la información que una neurona individual receipta, cerca del 10% proviene de impulsos exteriores y el 90% restante proviene de otras neuronas [13].

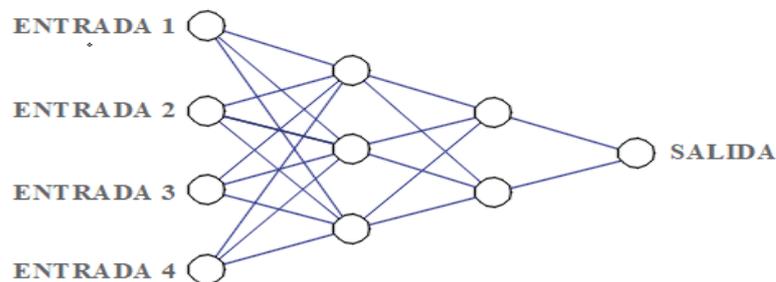
En la última década se han desarrollado procesadores computacionales que poseen capacidades de cálculo extraordinarias. Cálculos matemáticos largos y clasificaciones complejas, son trabajos que un ordenador puede realizar de forma sencilla en comparación al tiempo que un humano tardaría en finalizar estas labores. Para desactivarse o activarse un procesador computacional pueden necesitar no más de  $10^{-3}$  segundos, pero hay tareas que los humanos pueden realizar de una manera más eficaz que un ordenador como, reconocer una fruta o distinguir el color de una herramienta. Estos procesadores pueden realizar la matemática dentro de una red neuronal de manera sencilla, sin importar el número de cálculos que tenga que hacer.

En la Figura 3.1 las dendritas son las encargadas de recibir la información, el axón en conexión con estas últimas crea una relación de transferencia hacia las siguientes neuronas, logrando así un recorrido de la información.

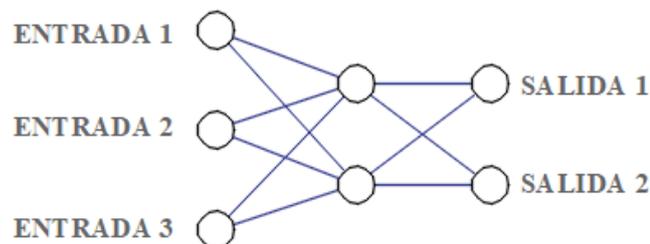


**Figura 3.1** Ilustración de una neurona biológica

Las Redes Neuronales (RN) y su teoría tienen como objetivo replicar este tipo de comportamiento. No modela las reacciones químicas del cerebro, tan solo se inspira en el flujo direccional que la información sigue en una neurona biológica.



**Figura 3.2** Red de múltiples entradas cuyos nodos son neuronas artificiales

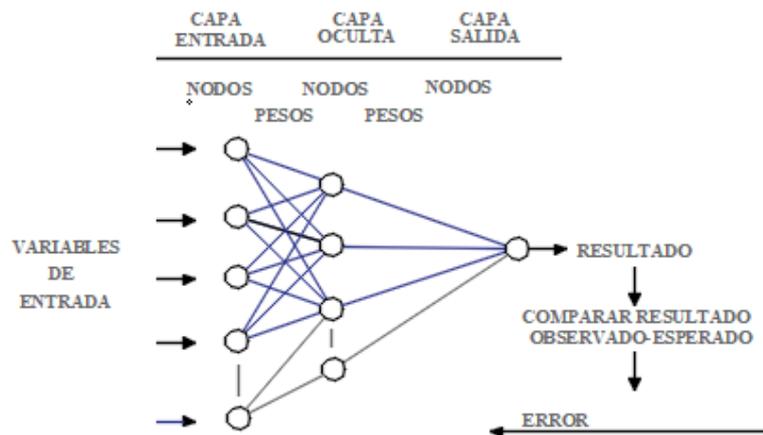


**Figura 3.3** Red de múltiples salidas cuyos nodos son neuronas artificiales

En las Figuras 3.2 y 3.3, cada neurona se conecta hacia la otra mediante arcos dirigidos, esto precisamente modela la conexión dendrita-axón.

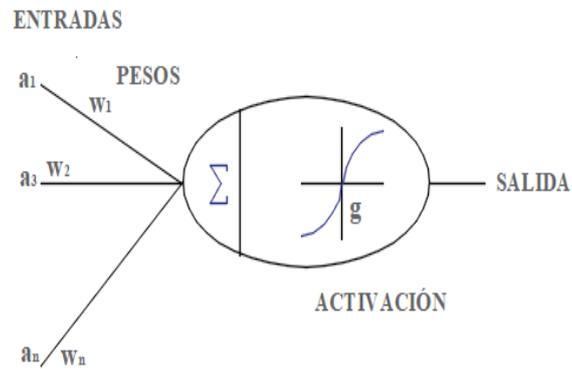
### 3.2 FUNDAMENTOS

Se puede entender a una red neuronal como un arreglo de capas de neuronas artificiales, distribuidas de tal manera que formen una estructura capaz de realizar los cálculos para la cual fue diseñada. Un conjunto de neuronas artificiales forma una capa de neuronas, así mismo, un conjunto de capas forma una red neuronal.



**Figura 3.4** Secciones de una red neuronal

Cada unidad o neurona artificial calcula su salida en función de las entradas que recibe. La salida de cada unidad puede servir a su vez como la entrada de otra. Una red recibe una serie de entradas externas y devuelve al exterior la salida de algunas de sus neuronas. El cálculo matemático que se realiza en cada unidad no es demasiado complejo.



**Figura 3.5** Ilustración de una neurona artificial

En las neuronas artificiales, cada arco tiene asociado un peso numérico notado como  $w_{ij}$  que determina la fuerza y el signo de la conexión (imitando la sinapsis).

La salida  $y_i$  de una neurona se calcula como:

$$y_i = g\left(\sum_{j=0}^n w_{ij} a_j\right) \quad (3.1)$$

Donde:

- $g$  es una función de activación
- La sumatoria  $(\sum_{j=0}^n w_{ij} a_j)$  se hace sobre todas las unidades  $j$  que envían su salida a la unidad  $i$ , esta se conoce como regla de propagación
- $w_{0i}$ : de cada unidad se interpreta como una cantidad que debe superar la suma de las señales de entrada que recibe la unidad, para que se active
- La función de activación  $g$  tiene el papel de normalizar la salida cuando el umbral de entrada se supera. Además, hace que la red no se comporte simplemente como una función lineal
- Si los índices  $i$  y  $j$  comienzan en 0 y se denota la entrada  $a = x$ , el peso umbral (bias)  $w_{i0} = \theta_i$  y  $x_0 = -1$ , podemos expresar:

Juan Fernando Segarra López

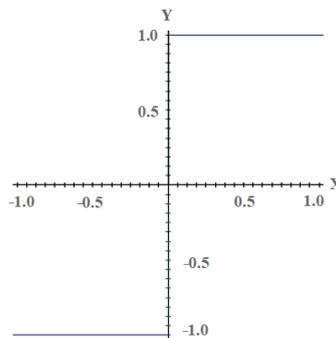
Pedro Sebastián Andrade Veloz

$$y_i = g(\sum_{j=1}^n w_{ij}x_j - \theta_i) \quad (3.2)$$

Existen varias funciones de activación que se usan para diferentes propósitos, siendo un criterio de selección el tipo de objetivo que se persiga; entre las funciones de activación más conocidas están las siguientes:

- La función bipolar (*sgn*)

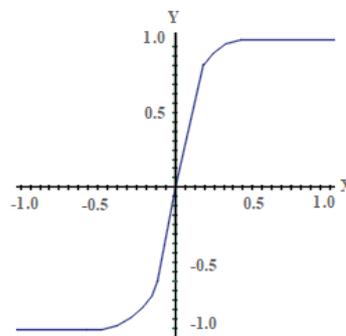
$$\text{sgn}(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x \leq 0 \end{cases} \quad (3.3)$$



**Figura 3.6** Gráfico cartesiano de la función bipolar

- La función tangente hiperbólica (*tanh*)

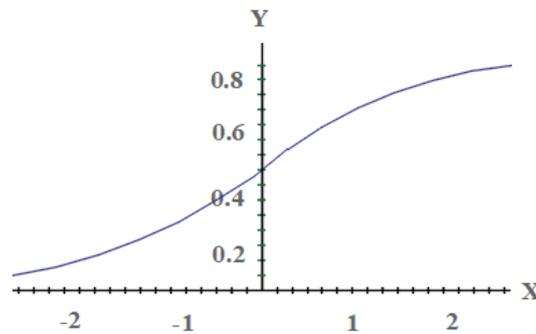
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.4)$$



**Figura 3.7** Gráfico cartesiano de la función *tanh*

- La función sigmoide ( $\sigma$ )

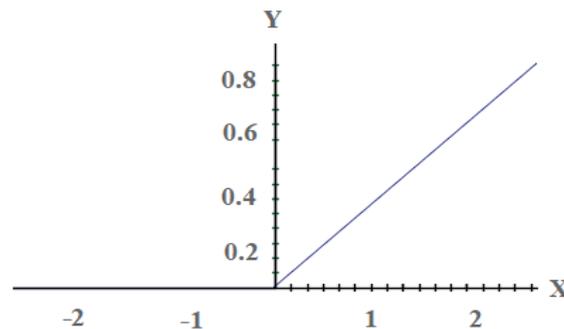
$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (3.5)$$



**Figura 3.8** Gráfico cartesiano de la función  $\sigma$

- La función unidad lineal rectificada (*ReLU*)

$$x^+ = \max(0, x) \quad (3.6)$$



**Figura 3.9** Gráfico cartesiano de la función *ReLU*

Por su uso en APM, la popularidad de la función *Relu* se ha incrementado; usualmente es usada en las capas de neuronas ocultas y muestra algunas desventajas en procesos de salida por su característica de rampa.

### 3.3 ARQUITECTURAS

Las RN tienen, según la forma de conexión y propiedades, diferentes tipos de arquitectura. Hebb (1949) postuló un mecanismo de aprendizaje, cuya idea básica consiste en que cuando un axón pre-sináptico causa la activación de cierta neurona post-sináptica, la eficacia de la sinapsis que las relaciona se refuerza. Es la razón por la cual se denomina aprendizaje Hebbiano a esta característica en la estructura de una neurona artificial.

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz

El fundamento matemático de las RN es extenso; no es el objetivo de este trabajo su estudio en detalle, sin embargo, se hará énfasis en los conceptos de vital importancia para la comprensión de cómo resolver el problema a tratar mediante el uso de una RN.

### 3.3.1. El perceptrón simple

El concepto introducido por Rosenblatt en 1962 muestra un sistema unidireccional compuesto por  $n$  entradas y  $m$  salidas. La operación para el cálculo de propagación de salida  $y_i$  de un perceptrón simple se expresa como:

$$y_i = g(\sum_{j=1}^n w_{ij}x_j - \theta_i) \quad \text{Con } i = 1, \dots, m \quad (3.7)$$

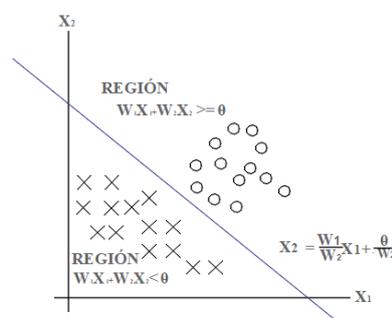
Larrañaga, Inza y Moujahid (2017) señalan que, con 2 entradas,  $x_1$  y  $x_2$  ubicadas en las abscisas y ordenadas respectivamente, se tiene:

$$y = \begin{cases} 1 & \text{si } w_1x_1 + w_2x_2 \geq \theta \\ 0 & \text{si } w_1x_1 + w_2x_2 < \theta \end{cases} \quad (3.8)$$

trabajando sobre  $x_2$

$$w_1x_1 + w_2x_2 - \theta = 0 \quad (3.9)$$

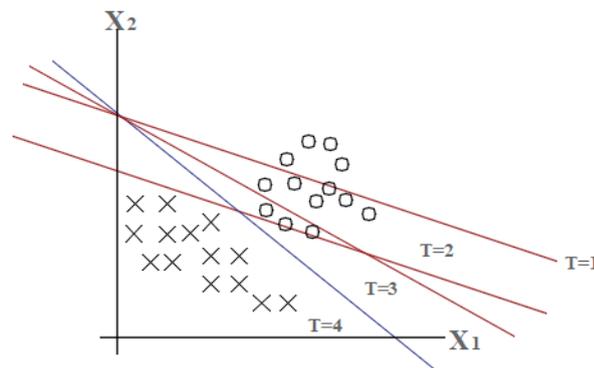
$$x_2 = \frac{w_1}{w_2}x_1 + \frac{\theta}{w_2} \quad (3.10)$$



**Figura 3.10** Región correspondiente a la decisión de un perceptrón simple

Este perceptrón se entiende como un clasificador. El algoritmo de Rosenblatt determina automáticamente los pesos sinápticos que clasifican patrones a partir de ejemplos etiquetados. Es conocido también como un discriminador lineal que ajusta el peso dependiendo del número  $t$  de iteraciones y los  $N$  pares de patrones entrada-salida según la fórmula:

$$w_{ij}(t+1) = w_{ij}(t) + \sum_{r=1}^N \Delta w_{ij}^r(t) \quad \text{Con } r = 1, \dots, N \quad (3.11)$$



**Figura 3.11** Ajuste y evolución de las decisiones de un perceptrón simple

Un perceptrón simple como discriminador puede clasificar datos dividiendo regiones mediante un hiper-plano o un plano como se observa en la Figura 3.10. Pero para tareas más complejas como divisiones no lineales existen herramientas que se han desarrollado tomando en cuenta este tipo de limitaciones.

### 3.3.2. El perceptrón multicapa

El perceptrón multicapa (PMC) introduce capas ocultas en una red, esto hace que varios de los problemas del perceptrón simple sean solucionados. Estos algoritmos usan para su entrenamiento un algoritmo de retro-propagación para actualizar sus pesos, de ahí que esta arquitectura se conozca también como redes de retro-propagación.

<<Werboz en 1974 introdujo por primera vez en su tesis doctoral la teoría del PMC, pero no fue hasta 1986 que Rumelhart lo popularizó gracias a que la computación había avanzado al punto de satisfacer los requisitos que el algoritmo de retro-propagación demandaba>> [14].

Considerando una capa oculta con función de activación sigmoide y una capa de salida con función de activación lineal, la salida  $z_k$  de la capa final de cada neurona  $k$  en una red PMC se calcula como:

$$z_k = \sum_{j=1}^o w'_{kj} y_j - \theta'_k = \sum_{j=1}^o w'_{kj} f\left(\sum_{i=1}^n w'_{ki} x_j - \theta_j\right) - \theta'_k \quad (3.12)$$

Los diferentes  $\theta_j$  representarán los umbrales correspondientes a la capa oculta,  $w'_{kj}$  los pesos de la capa de salida y  $\theta'_k$  sus umbrales respectivos.

Dado un patrón de entrada  $x^r$  ( $r = 1, \dots, N$ ) se puede generalizar la ecuación 3.12 para cualquier número  $k$  de neuronas de salida:

$$z^r_k = \sum_{j=1}^o w'_{kj} y_j - \theta'_k = \sum_{j=1}^o w'_{kj} f\left(\sum_{i=1}^n w'_{ji} x^r_j - \theta_j\right) - \theta'_k \quad (3.13)$$

Adicional a esto, la arquitectura PMC, adiciona 2 procesos, el primero consiste en la regla de aprendizaje, usando la regla de los mínimos cuadrados o LSM por su inglés, (en inglés Least Mean Square) y un optimizador basado en la regla LSM conocido como el descenso por el gradiente. El gradiente  $\nabla E(w)$  a su vez es la derivada parcial del error con respecto al peso y es usado para encontrar una configuración de pesos óptima hasta que el error sea mínimo. Matemáticamente se recorre la hiper-superficie del error hasta encontrar un mínimo local, el proceso es iterativo y se expresa como:

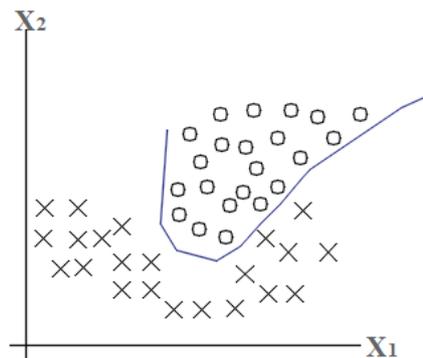
$$w(t+1) = w(t) - \eta \nabla E(w) \quad (3.14)$$

Para iniciar el algoritmo se elige un valor aleatorio de pesos y umbrales iniciales ( $t=0$ ). La constante  $\eta$  es positiva y pequeña, llamada factor de aprendizaje, modera la actualización de pesos. Se debe iniciar con pesos pequeños, positivos y negativos, se debe elegir también la variación de pesos para cada conjunto de entrenamiento, este proceso se conoce como aprendizaje por lotes.

<<El algoritmo de retro-propagación se puede justificar formalmente como un descenso por el gradiente del error, la salida de cada unidad se calcula propagando valores hacia adelante, pero el error en cada una se calcula desde la capa de salida hacia atrás (de ahí el nombre de retro-propagación)>> [13].

Un conjunto de entrenamiento  $D$  debe tener ejemplos de la forma  $(\vec{x}, \vec{y})$ , con  $\vec{x} \in R^n$  y  $\vec{y} \in R^m$  ( $n$  y  $m$ , dimensión de la entrada y salida respectivamente), un factor de aprendizaje  $\eta$ , una función de activación  $g$  diferenciable y una estructura de red.

De forma gráfica, como se observa en la Figura 3.12, esta arquitectura puede realizar separaciones complejas a diferencia de un perceptrón simple, por ejemplo, la función XOR, ya que a diferencia de la función AND y OR, no es linealmente separable.



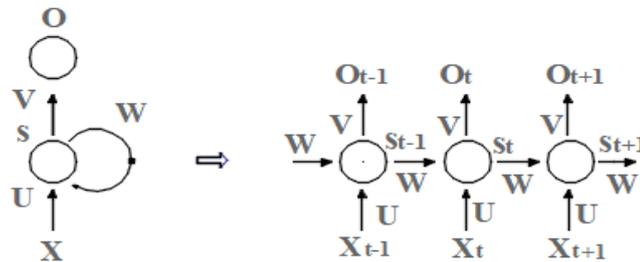
**Figura 3.12** Región de decisión arbitraria de una RN con arquitectura PMC

### 3.3.3. Redes neuronales recurrentes

Si bien las redes PMC pueden discriminar situaciones no lineales, existen tareas dificultosas para esta arquitectura, como permitir que una salida alimente a las entradas.

Mejorados con la conexión completa y recurrente de la red, los algoritmos que forman una red neuronal recurrente (RNR) satisfacen muchas de las

problemáticas presentes en arquitecturas más simples. El propósito de una RNR, es otorgar una suerte de memoria a la red.



**Figura 3.13** Una RNN y el despliegue de su cálculo en el tiempo

- $x_t$  es la entrada en el tiempo  $t$ , por ejemplo  $x_1$  puede ser un vector correspondiente a la segunda palabra de una oración
- $s_t$  es el estado oculto en el tiempo  $t$  es calculado en base a los estados ocultos previos y en la entrada en el justo momento:  $s(t) = (Ux_t - Ws_{t-1})$
- $o_t$  es la salida en el tiempo  $t$ . Por ejemplo, si queremos predecir la siguiente palabra de una oración, será un vector de probabilidades a través de un conjunto de entrenamiento.  $o_t = softmax(U_{s_t})$  [15]

Una RNR debe ser debidamente diseñada y entrenada antes de poder ejecutarse. El entrenamiento se da sobre un conjunto de datos referentes al problema. Por su parte, el diseño considera una amplia selección de hiperparámetros tales como: el porcentaje de datos usados para el entrenamiento, porcentaje de datos usados para la validación, función de activación, número de iteraciones o épocas, aprendizaje por lotes, etc.

Un ejemplo ampliamente usado para mostrar el funcionamiento de una RNR es el de predecir la última palabra de una oración como la siguiente:

“Mi amigo francés, vive en París, mi amigo habla fluidamente el idioma \_\_\_\_\_”



Si el diseño y el entrenamiento de una RNR son debidamente ajustados, la red podrá predecir la palabra francés, por lo tanto, encontrará la solución óptima al problema.

Ahora, teniendo en cuenta la eficacia de estas redes para problemas de este tipo, existen inconvenientes con las dependencias a largo plazo, para esto, el mismo ejemplo anterior puede ser reformulado como sigue:

*“Mi amigo francés, vive en París, su esposa es de origen inglés y su mascota es un cachorro de pastor alemán, mi amigo habla fluidamente el idioma \_\_\_\_\_”*

Si bien la lógica es la misma, la respuesta no es tan sencilla para una RNR, dado que existe una gran cantidad de datos considerados como idiomas, el algoritmo acertadamente otorgará pesos importantes a estos y la salida no es tan intuitiva como antes, puesto que se puede perder la relevancia numérica del peso de un dato antiguo.

La palabra francés, por ser el dato más antiguo en la oración (vista desde el final del texto hacia el principio), puede presentar una desaparición con respecto al valor de su gradiente. A este problema matemáticamente se lo denomina como el problema de desaparición del gradiente. Este fenómeno tiene su inverso llamado el problema de expansión del gradiente.

Para estos problemas de dependencias de largo plazo se han desarrollado soluciones modificando la misma arquitectura RNR y desarrollando un tipo especial de la misma, conocido como redes de memoria de largo plazo.

#### **3.3.4. Redes de memoria de largo plazo**

Para familiarizarse con las redes de memoria de largo plazo (RMLP), no hace falta más que entender lo que es la persistencia en el pensamiento. El hombre independiente del género o edad, razona cualquier cosa partiendo del conocimiento adquirido. Un niño no puede cantar en un recital si aún no aprende a hablar y es

porque el pensamiento humano tiene persistencia en la memoria, y es justamente lo que este tipo de redes persiguen.

Las RMLP son muy populares en la actualidad por sus aplicaciones en pronósticos y reconocimientos, no poseen una diferencia fundamental de las RNR, pero se distinguen en los cálculos que realizan para el almacenamiento de memoria en la red.

La memoria en las RMLP puede entenderse como una caja negra donde se procesa el estado oculto anterior ( $h_{t-1}$ ) y la entrada actual ( $x_t$ ) de una neurona, cargando la responsabilidad de elegir lo que se conservará para la propagación de datos o lo que se eliminará en el paso a la siguiente capa de neuronas.

En [16] se expone el cálculo para una red RMLP de una manera completa, pero sin incurrir en exhaustiva complejidad. Define el área de toma de decisiones con respecto a la memoria como celdas con un respectivo estado.

La primera etapa del cálculo consiste en decidir que se suprime en el estado de celda, la decisión se realiza mediante una capa sigmoide llamada puerto de olvido, que analiza  $h_{t-1}$  y  $x_t$ , con la adición de un nuevo umbral ( $b_f$ ) y produce una salida entre 0 y 1 para cada número en el estado de celda  $C_{t-1}$ . El 0 significa olvidar completamente la característica, y el 1 mantener completamente la característica.

$$f(t) = \sigma(w_t[h_{t-1}, x_t] + b_f) \quad (3.15)$$

En la segunda etapa, se decide lo que se va a almacenar en la celda, primero una capa de función  $\sigma$ , llamada puerto de entrada, toma en cuenta solo los valores se van a actualizar, para que una capa con función  $\tanh$  cree un vector con valores nuevos  $\bar{C}_t$  que serán añadidos al estado actual de la celda.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (3.16)$$

$$\bar{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \quad (3.17)$$

Ahora que los dos puertos, el de entrada y el de olvido, han tomado las decisiones de lo que se va a conservar y de lo que no, se deben cumplir estas decisiones. Actualizando el anterior estado de la celda  $C_{t-1}$  a uno nuevo  $C_t$ , se multiplica el antiguo estado por  $f(t)$  (olvidando lo que se decidió olvidar) se añade  $i_t * \bar{C}_t$ . Estos son los nuevos valores.

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \quad (3.18)$$

Por último, se decide cuál va a ser la salida, esta decisión estará gobernada por el estado de la celda, se ejecuta una capa sigmoide que considera que parte de la celda va a mostrarse en la salida. Luego una función  $\tanh$  coloca a los valores en un rango entre -1 y 1, y después lo multiplica por la salida de la capa sigmoide, así solo se muestra como salida la parte deseada.

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3.19)$$

$$h_t = O_t * \tanh(C_t) \quad (3.20)$$

Por la eficacia mostrada para guardar valores relevantes y persistencia en la memoria, las RMLP han sido usadas ampliamente para realizar predicciones y reconocimientos de todo tipo. Por lo que encontrará la solución al problema:

*“Mi amigo francés, vive en París, su esposa es de origen inglés y su mascota es un cachorro de pastor alemán, mi amigo habla fluidamente el idioma francés”*

## 3.4 APRENDIZAJE AUTOMÁTICO

### 3.4.1 Aprendizaje de máquina



<<Un programa de computadora aprende de la experiencia (E) con respecto a alguna clase de tareas (T) y medida de desempeño (P), si su desempeño en tareas en (T), medido por (P), mejora con la experiencia (E)>> [17].

El aprendizaje de maquina está tan relacionado a la minería de datos y a la estadística como a la inteligencia artificial y es que no solo trata de aprendizaje, sino de representación de memoria, toma de decisiones e incluso de análisis abstracto de información que está oculta; entonces, si un programa mejora su desempeño en una tarea basado en experiencia pasada, el programa aprendió.

Globalmente se pueden agrupar a los AAM en grupos donde fácilmente pueden encajar según sus funciones y características, distinguiendo 3 principales grupos.

- i. El primer grupo de algoritmos son llamados los algoritmos de aprendizaje supervisado, donde el algoritmo intenta producir una salida tomando en cuenta etiquetas de las entradas, un ejemplo común es el agrupamiento selectivo, más conocido por su nombre en inglés Clustering.
- ii. El aprendizaje no supervisado es el segundo grupo de la clasificación de estos, al no tener etiquetas de entradas el propio algoritmo debe ser capaz de categorizar las mismas, con el fin de producir salidas en base a entradas con patrones encontrados.
- iii. El tercer grupo de estos algoritmos es el de aprendizaje semi-supervisado, combina los dos anteriores, tienen en cuenta entradas etiquetadas y las no etiquetadas para poder clasificar de manera adecuada.

### **3.4.2 Aprendizaje profundo de máquina (APM)**

El APM puede ser considerado un tipo especial de los AAM. Teóricamente fue desarrollado para que los AAM alcancen un objetivo muy especial, la total



independencia (no supervisión) en la toma de decisiones de un proceso o análisis. Pero no siempre los algoritmos de APM son no supervisados.

El APM mundialmente conocido como Deep Learning es entonces solo una rama concreta dentro del Machine Learning (Aprendizaje de Máquina), que agrupa algoritmos conexionistas, los más famosos son las redes neuronales, que como se estudió en el capítulo anterior del presente, algunas arquitecturas pueden lograr la propagación de datos (hacia adelante y recurrentes<<acción de que algo vuelva a aparecer>>) y la independencia buscada.

Compañías como Apple usan APM, en su asistente virtual SIRI, lo mismo hace Microsoft y Google con Cortana y Google Now respectivamente, aprovechando este tipo de programas para entre otras cosas, realizar llamadas telefónicas mediante reconocimiento de voz o programar alarmas con los hábitos del usuario [18]. Empresas y grupos de la bolsa de valores a nivel mundial dan un trato más sensible desde el punto de vista económico al APM, si bien la bolsa de valores tiene características muy aleatorias e imposibles de predecir, todas estas compañías, entre ellas las entidades financieras como Morgan Stanley y Citigroup, contratan expertos de AAM para sus oficinas de riesgo, de hecho, cerca del 70% de ordenes en Wall Street están comandadas por software [19].

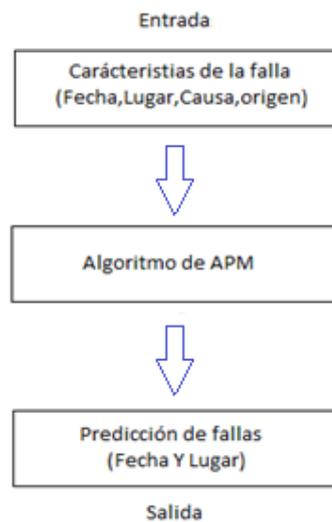
La relevancia y el exponencial crecimiento que tiene el APM en el mundo entero, es muestra fidedigna que la mayoría de problemas relacionados con la previsión, análisis y predicción de datos, sin importar su índole son atacados con esta herramienta.

### **3.5 HERRAMIENTAS DE SOFTWARE USADAS PARA LA REDES NEURONALES APLICADAS A LA PREDICCIÓN DE FALLAS**

Las RN, sobre todo las RMLP y la actual facilidad de implementación de estas mediante la librería TensorFlow y su módulo Keras en Python ofrecen una solución matemática muy superior a la estadística convencional. Los manuales de mantenimiento preventivo, suposiciones por experiencia laboral o cálculos aleatorios

de ocurrencia de fallas son métodos inexactos, el algoritmo propuesto es una combinación educada estadísticamente de los elementos más importantes de todo lo anteriormente mencionado.

Información como la ubicación, el origen, la fecha y el tipo de una falla, son características persistentes a lo largo del tiempo, haciendo que una predicción convencional sea difícil. Pero no es una condición en la que una red RMLP no pueda desenvolverse de manera correcta.



**Figura 3.14** Esquema general del modelo computacional propuesto

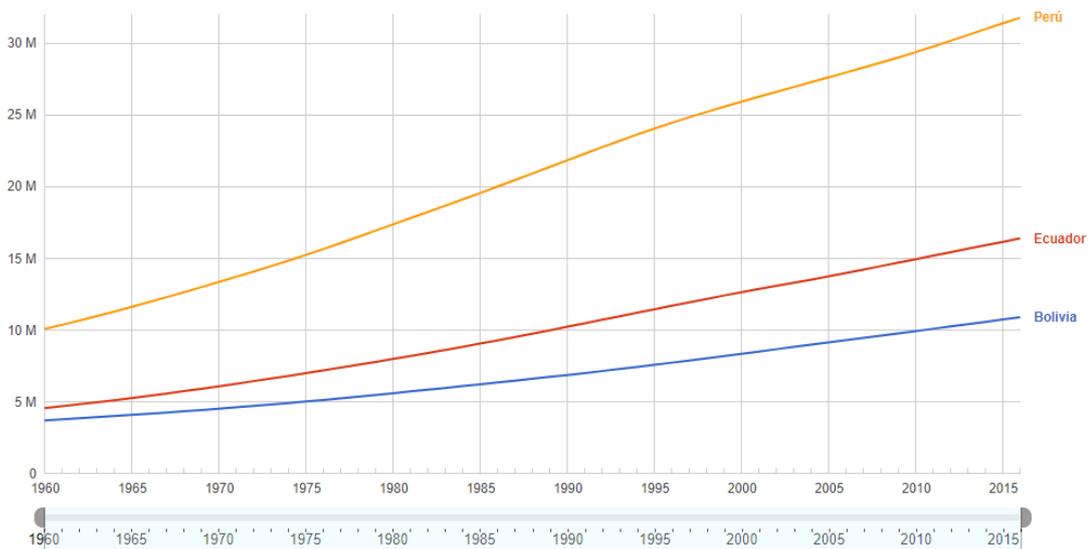
## CAPÍTULO 4

### 4 APLICACIÓN DEL MODELO

#### 4.1 INTRODUCCIÓN

La matemática en general facilita la comprensión de cualquier planteamiento científico. Sucesos en el tiempo pueden ser representados por gráficas cartesianas, donde el eje de las abscisas contiene información numérica relacionada a un evento y el eje de las ordenadas representa el tiempo.

Las series temporales son usadas en casi todas las áreas de la ciencia. Exactas o no, representan un método analítico muy preciso para el estudio de diferentes datos. Es así como las ciencias sociales pueden modelar el incremento de la población mundial con el paso de los años.



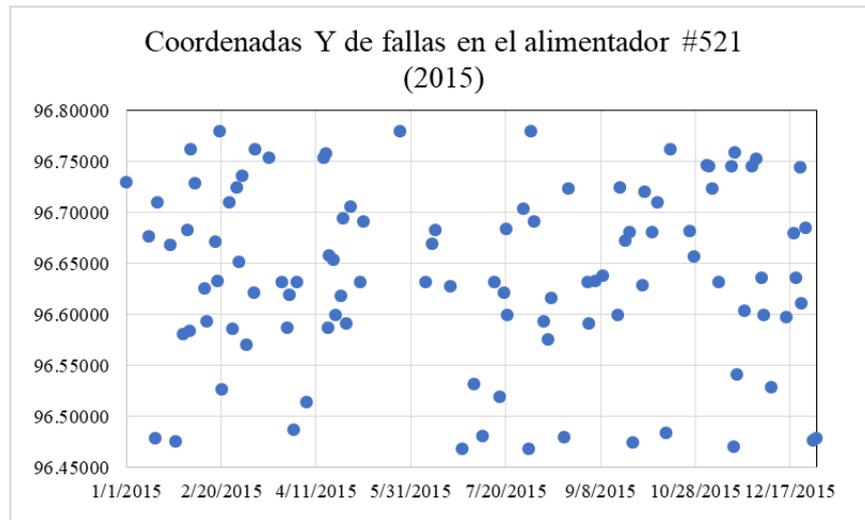
**Figura 4.1** Incremento poblacional de Ecuador, Perú y Bolivia [21]

Este tipo de gráficas en dos dimensiones pueden modelar aparte de series temporales, relaciones de magnitudes. En el caso de la Física, se puede modelar el incremento de la velocidad con respecto a la aceleración o el movimiento oscilatorio de un péndulo.

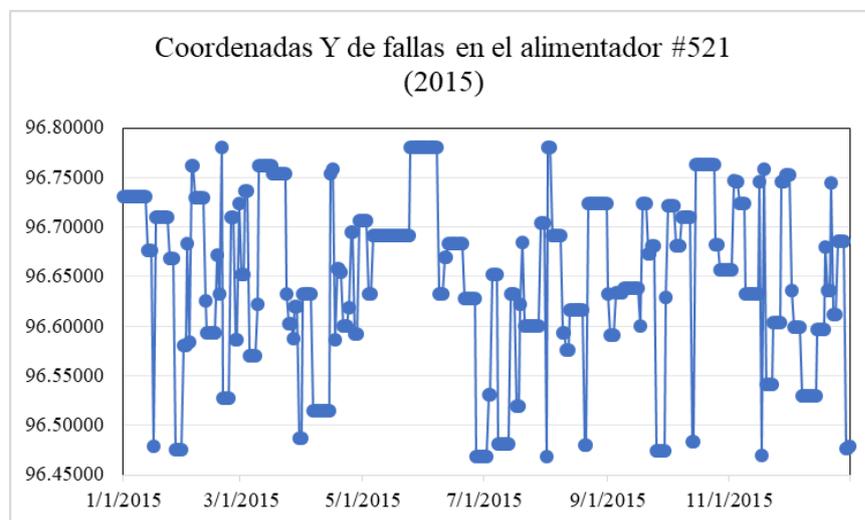
Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz

Considerando el amplio rango de aplicaciones que ofrece estas curvas, es posible modelar también la ocurrencia de fallas en un alimentador primario de la CENTROSUR con respecto a los meses del año. El eje de las ordenadas mostrará la ubicación de la falla y eje de las abscisas la fecha cuando sucedió.



**Figura 4.2** Datos reales de fallas en un alimentador



**Figura 4.3** Aproximación mediante series de tiempo de las fallas en un alimentador

Las coordenadas UTM (Universal Transversa Mercator), presentan un sistema que en espacios pequeños a nivel global se considera plano. Las coordenadas planas (x, y) son el par ordenado que indican la latitud y longitud

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz



georreferenciada de un punto. Es por esto que en una aproximación por series de tiempo se puede modelar la falla en un alimentador, donde el eje de las ordenadas adopta el valor las coordenadas X o Y de una falla y el eje de las abscisas el tiempo [22].

En el Ecuador, la provincia del Azuay en la región sierra está ubicada aproximadamente en coordenadas Planas UTM:

Norte: 9668240 / 9686630 y Este: 694500 / 722330 [23].

La solución que se plantea al problema radica en, con base a un gráfico de coordenadas bidimensional predecir su comportamiento en los siguientes intervalos de tiempo mediante un método computacional basado en APM. Para esto se usará las características que la curva presenta en un tiempo anterior a la predicción que se desea obtener. Finalmente, la predicción del comportamiento de la curva resulta en un indicativo de dónde y cuándo una falla pueda generarse.

## 4.2 DESCRIPCIÓN DEL MODELO

El modelo en sí, se puede resumir en 6 etapas:

- i. Ordenar los datos de la CENTROSUR en una hoja de cálculo (archivo de extensión .csv <<valores separados por coma “comma-separated values” >>).
- ii. Leer los datos de entrada en un .csv con pandas y normalizar el conjunto de datos usando sklearn para colocarlos en un vector mediante numpy
- iii. Iniciar una red RMLP o LSTM (por su inglés “long-short term memory”)
- iv. Calcular el error y reajustar los hiper-parámetros hasta minimizarlo
- v. Obtener las predicciones y des-normalizarlas para obtener resultados legibles

vi. Imprimir las predicciones o graficar los resultados con Mathplotlib

Estas etapas se pueden observar mediante el siguiente diagrama de flujo:

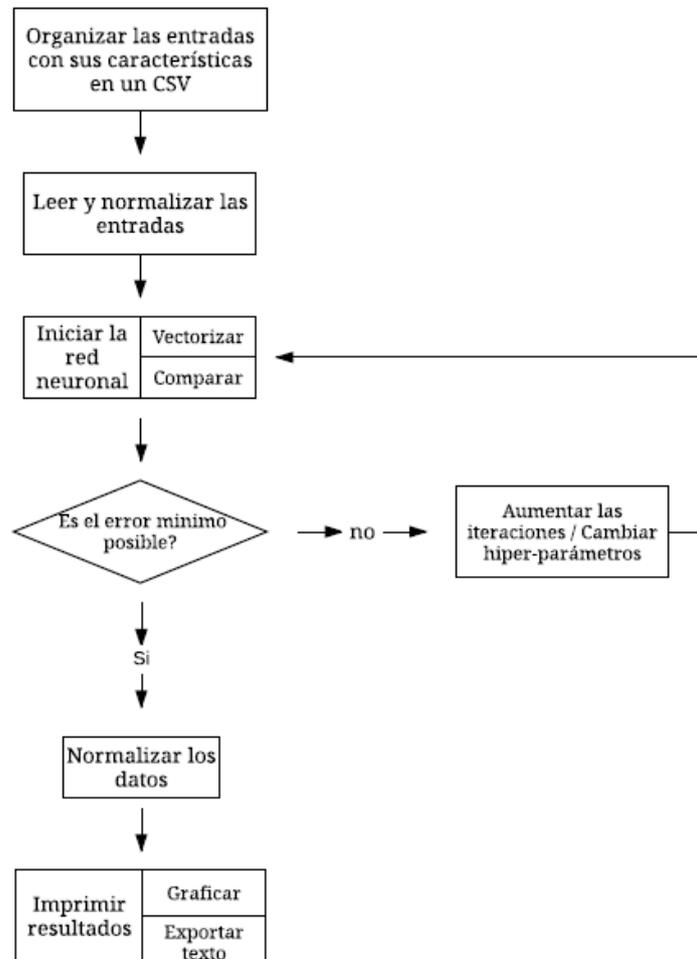


Figura 4.4 Diagrama de flujo del modelo propuesto

## 4.3 PROTOTIPO DEL MODELO PROPUESTO

### 4.3.1 Modelo para una característica

Como entrada se eligió una columna (característica) con quinientas treinta y dos filas de números que gobiernan el comportamiento repetitivo de la función Seno.

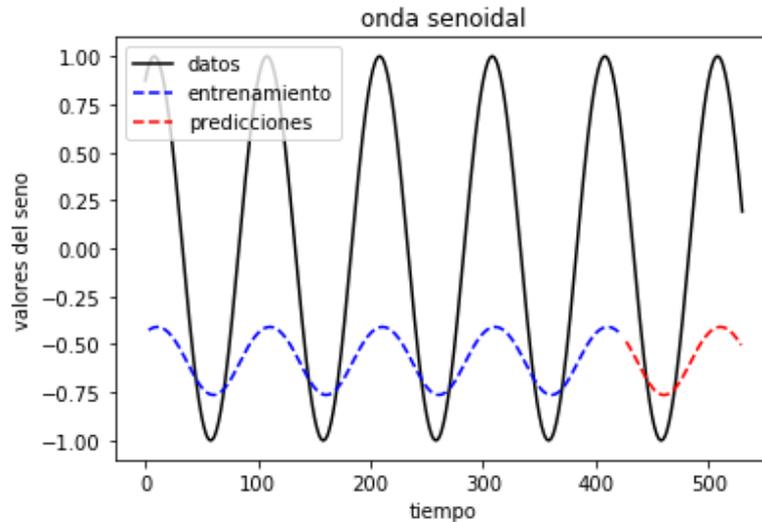


Una onda senoidal es aritméticamente oscilante entre -1 y 1, con una periodicidad marcada, por lo que se tomó como objetivo en las primeras versiones del algoritmo intentar predecir el comportamiento de esta sencilla secuencia numérica.

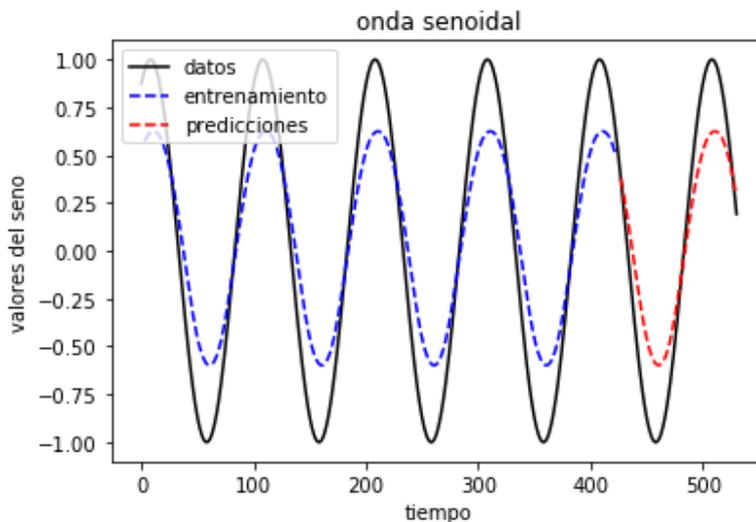
En cada una de las pruebas se modificó el número total de iteraciones (*epochs*), logrando diferentes resultados mostrados desde la Figura 4.5 hasta la Figura 4.8. En las mismas se observó que a mayor número *epoch* mejor se realiza la predicción, pero el costo computacional entendido como tiempo, crecía.

Para todas las pruebas se eligió un valor típicamente usado del 70% de los datos para el entrenamiento y el 30% para pruebas [24]. En cuanto a la configuración de la red RMLP se usó 2 capas clásicas, la de entrada y la de salida, mostrando resultados satisfactorios como se verá a continuación (la influencia del número de capas ocultas se estudia a fondo en el modelo de varias características, porque en este prototipo la configuración más simple mostró un desempeño sobresaliente).

La Figura 4.6 muestra una mejoría notable de la predicción con respecto a la Figura 4.5, pero considerando que las iteraciones aumentan en dos, de manera prematura se puede concluir que las predicciones (en rojo) progresan de forma extraordinaria con tan solo dos iteraciones adicionales.



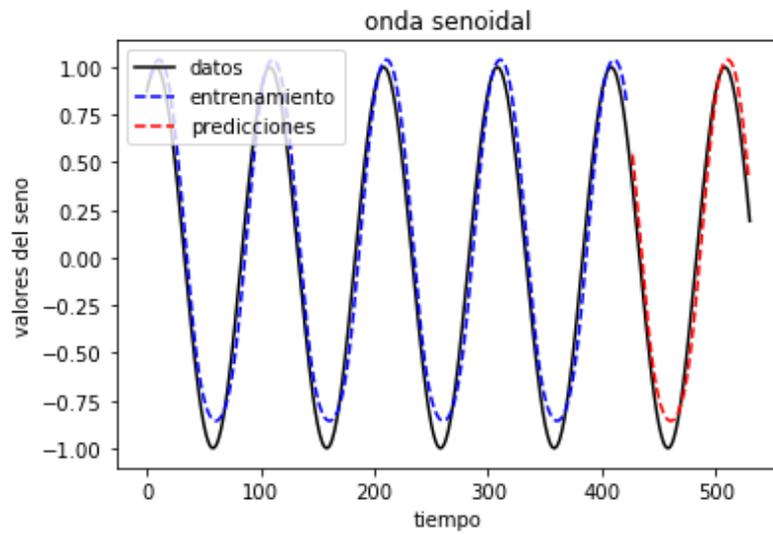
**Figura 4.5** Predicción de una onda senoidal, *epochs* = 2



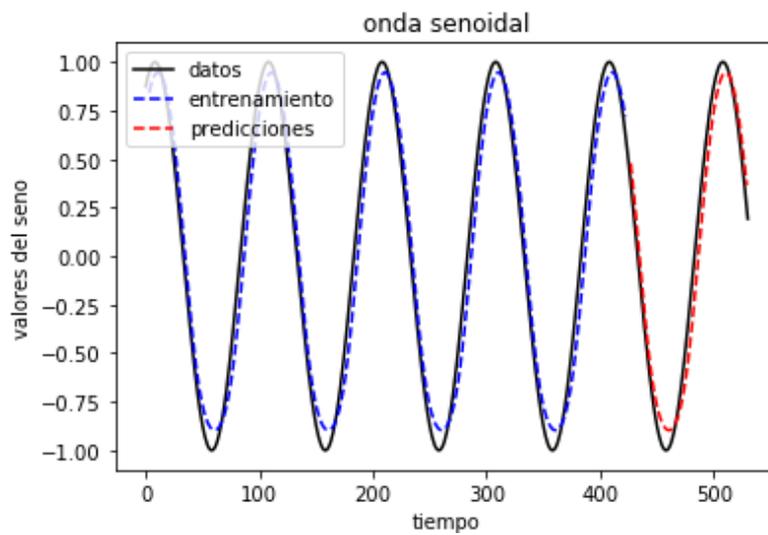
**Figura 4.6** Predicción de una onda senoidal, *epochs* = 4

Ahora, después de analizar las Figuras 4.7 y 4.8, se muestra que a diferencia de la anterior comparación, aumentar los *epochs* a 8 y después a 16, no refleja nada concluyente.

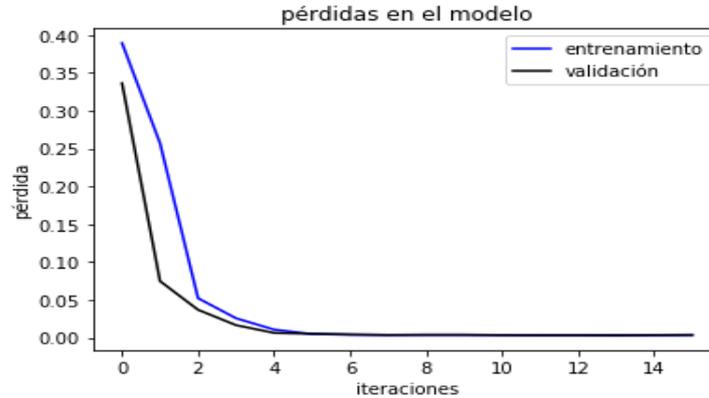
Las pérdidas se calculan como el error tras cada *epoch*, y es igual a la raíz cuadrada del error cuadrático medio (RMSE 'siglas de Root Mean Squared Error') para visualizar mejor lo que pasa en la red se graficaron las pérdidas con respecto a las iteraciones. Esto último puede ser observado en la Figura 4.9.



**Figura 4.7** Predicción de una onda senoidal, *epochs* = 8



**Figura 4.8** Predicción de una onda senoidal, *epochs* = 16



**Figura 4. 9** Pérdidas vs las iteraciones en el modelo de APM

Finalmente, se concluye que el modelo para una característica tiene total validez a la hora de predecir una curva senoidal y una predicción correcta se puede considerar óptima a partir de las ocho iteraciones, y más allá de esto, la mejoría es despreciable y no tendría utilidad gastar recursos incrementado el *epoch*.

#### 4.4 MODELO PARA LA PREDICCIÓN DE FALLAS EN UN ALIMENTADOR

##### 4.4.1 Modelo para varias características

Al probar la eficiencia del algoritmo en una serie de tiempo como la función seno, se debe tener en cuenta que la entrada en dicho algoritmo es una columna simple de números. Por lo tanto, resulta en un desafío menor al de un sistema con varias entradas en columnas para características diferentes.

Si la periodicidad está demasiado marcada, y la solución es evidente, con estadística básica podría resolverse el problema. Como la realidad no puede estar más alejada de esto, las soluciones triviales en este caso son inviables. En un problema donde no existe una reproducibilidad tan sencilla como la que existía en una onda senoidal, un modelo computacional basado en APM puede ayudar a resolver el problema.



Por todo lo anterior, se ha desarrollado un método complejo de redes neuronales, para de una manera óptima producir predicciones razonables, muy difíciles de obtener mediante herramientas convencionales.

Partiendo como base el algoritmo de predicción de una onda senoidal, se alteraron varios parámetros y cambiaron algunas consideraciones en la red:

- i. La entrada cambio: de una columna con quinientos treinta y dos filas, a veinte columnas con trescientos sesenta y cinco filas
- ii. La red, ahora presenta varias versiones según el número de capas ocultas para un posterior análisis
- iii. La función de activación en la salida se modificó en varias pruebas
- iv. El tamaño del lote o *batch* (limitador del número de ejemplos de datos que se mostrarán a la red antes de poder realizar una actualización de peso) se varió según las pruebas.

Se usó específicamente como entrada las siguientes características de una falla: fecha de desconexión (año, mes y día), causa, coordenada  $X$  y coordenada  $Y$ . Fijando como objetivo la obtención de las coordenadas  $X$  e  $Y$  en intervalos de tiempo futuros. A comparación del algoritmo prototipo, donde se trabajó con una sola característica de los datos.

Los datos escogidos como entrada de este modelo fueron aquellos que la empresa tenía mejor estructurados y completos. Se tomaron en cuenta las entradas desde el 2011 hasta el 2015 de las fallas en el alimentador #521. Produciendo trescientos sesenta y cinco (días del año) filas con 20 columnas (cuatro columnas por año, una en base a cada característica de la falla en dicho alimentador).

Muchas de las características de las fallas como las fechas y coordenadas en 2016 y 2010 del alimentador #521 mostraban inconsistencias en algunos meses. A

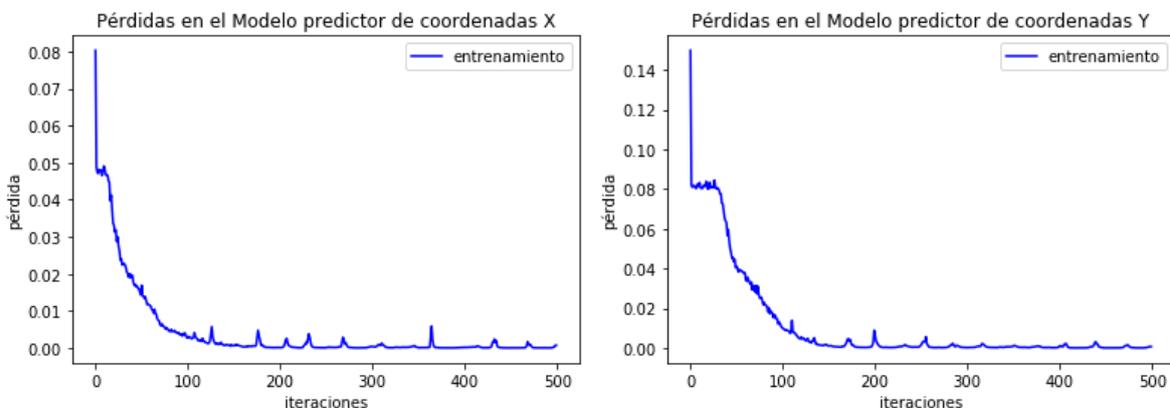
Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz

menudo en estos casos la empresa registró las fallas propias de diciembre e incluso noviembre para enero del siguiente año, siendo esto una imprecisión. Además, se encontró la carencia absoluta de características en algunos alimentadores. El registro de información histórica del último mantenimiento realizado en los alimentadores estaba incompleto.

En la primera prueba se consideró *epochs* al azar igual a 500, y una configuración típica: *train\_size* (*entrenamiento*) = 70% y *test\_size* (*pruebas*) = 30% y un 10% para validación para una red sin capas ocultas (partiendo del caso más simple).

El aporte más valioso del resultado mostrado en la Figura 4.9 es que a partir de cierto punto donde el error se vuelve constante, aumentar los *epochs* es innecesario. Un vistazo a la Figura 4.10 revela que aproximadamente con ciento cincuenta iteraciones el modelo deja de presentar pérdidas significativas en ambos casos (predicción de Coordenada X e Y).



**Figura 4.10** Pérdidas versus iteraciones en el modelo de varias características.

El parámetro *batch* es muy influyente en los problemas de predicción, este parámetro influye directamente en la predicción que se va a realizar en cada intervalo de tiempo  $t$ , por ejemplo: un programa de ajedrez está obligatoriamente forzado a usar un *batch* alto puesto que las predicciones que realizará se verán mejoradas si consideran varias jugadas futuras.



Por otro lado, una serie temporal se verá afectada si se usa un *batch* alto, ya que para estas es mejor considerar las predicciones para cada punto siguiente inmediato al anterior (algo que un jugador avanzado de ajedrez nunca haría).

Con el número *epoch* fijado como base en 150 se tomó en consideración ampliar el número de capas ocultas en la red para observar el comportamiento de esta, los diferentes resultados se muestran a continuación.

#### 4.4.1.1 Modelo sin capas ocultas

##### 4.4.1.1.1 Variación del hiper parámetro *batch* sin capas ocultas

En las Figuras 4.11 - 4.14, se muestran los resultados de la predicción en una red sin capas ocultas en base a la variación del *batch*. Las Tablas 4.1 y 4.2 muestran el error correspondiente a cada una de estas variaciones, concluyendo que el error mínimo se da con  $batch = 2$ , ratificando esto que el *batch* más pequeño es el más conveniente en el ajuste de hiper parámetros para un modelo de predicción de series de tiempo.

Consideraciones:

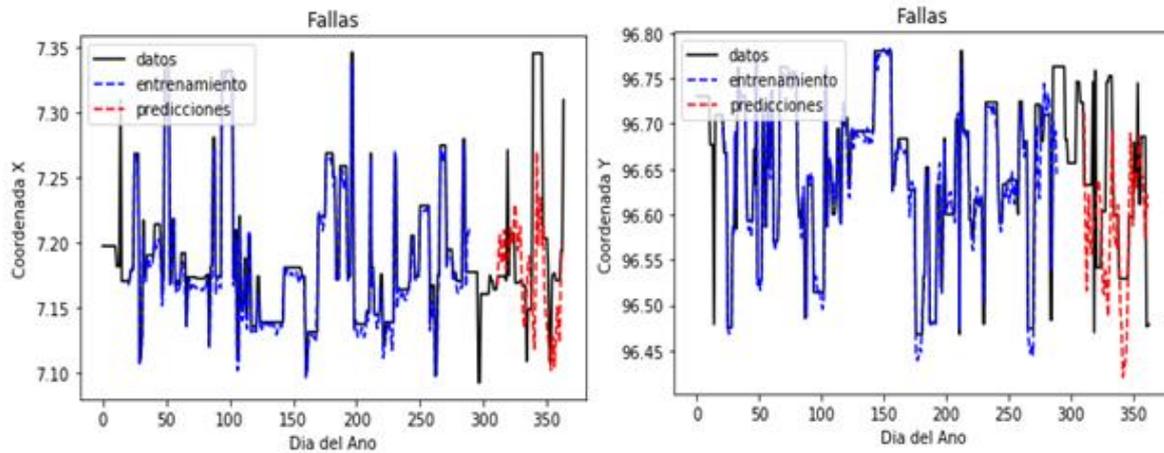
- i. RMSE Entrenamiento: Es la raíz cuadrada del error cuadrático medio presentado por la red durante el entrenamiento después de 150 iteraciones
- ii. RMSE Pruebas: Es la raíz cuadrada del error cuadrático medio presentado por la red durante las pruebas después de 150 iteraciones
- iii. Mientras mayores sean estos errores, la predicción es más inexacta y el modelo puede estar al borde de la sobre-alimentación, esto es más fácil visualizarlo en las gráficas siguientes.

**Tabla 4.1** Error de una red con una capa oculta para diferentes *batch*, predicción de coordenadas X.

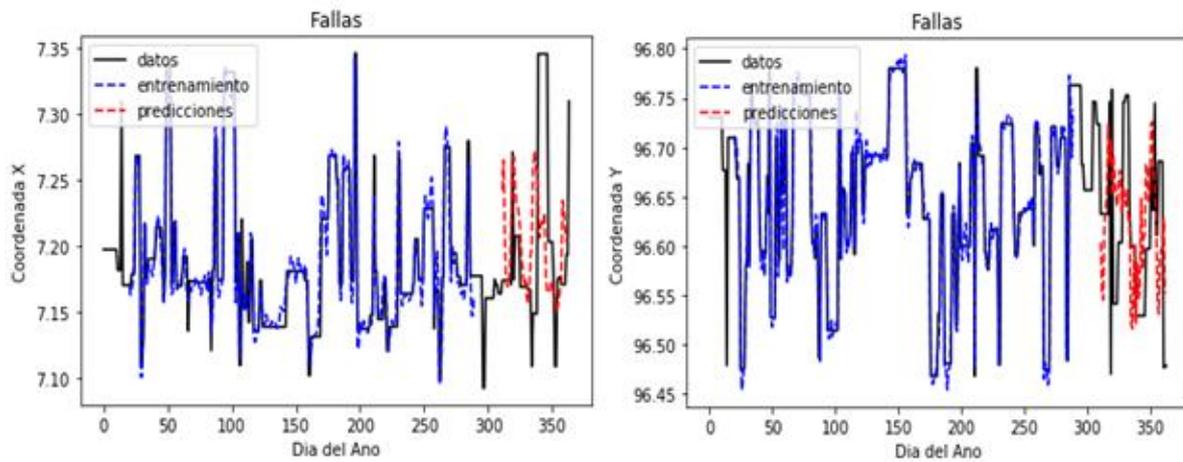
| Coordenada X      | RMSE Entrenamiento | RMSE Pruebas |
|-------------------|--------------------|--------------|
| Capas Ocultas     | 0                  | 0            |
| <i>Batch</i> = 2  | 0.01               | 0.07         |
| <i>Batch</i> = 5  | 0.01               | 0.07         |
| <i>Batch</i> = 10 | 0.02               | 0.08         |
| <i>Batch</i> = 15 | 0.01               | 0.07         |

**Tabla 4.2** Error de una red con una capa oculta para diferentes *batch*, predicción de coordenadas Y.

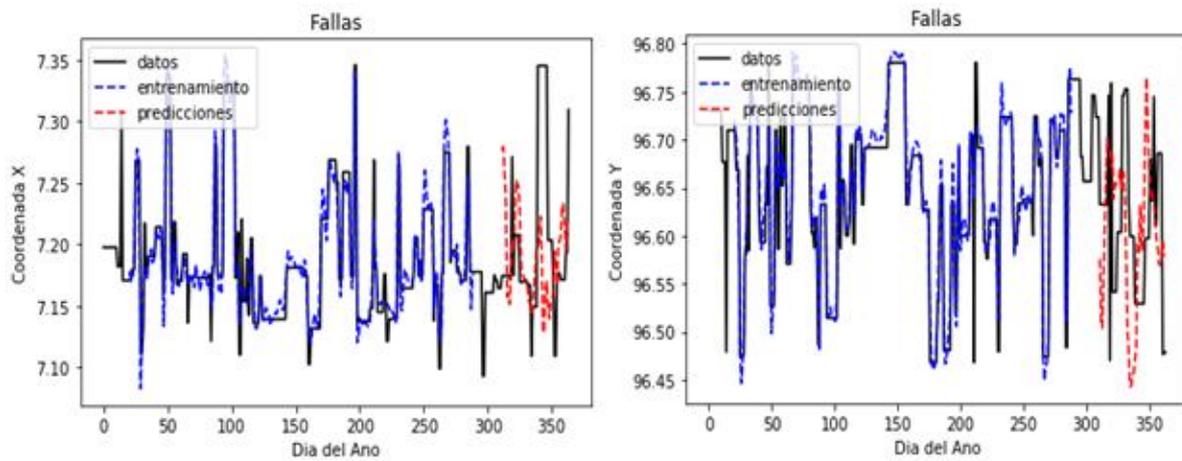
| Coordenada Y      | RMSE Entrenamiento | RMSE Pruebas |
|-------------------|--------------------|--------------|
| Capas Ocultas     | 0                  | 0            |
| <i>Batch</i> = 2  | 0.02               | 0.1          |
| <i>Batch</i> = 5  | 0.01               | 0.09         |
| <i>Batch</i> = 10 | 0.03               | 0.11         |
| <i>Batch</i> = 15 | 0.03               | 0.09         |



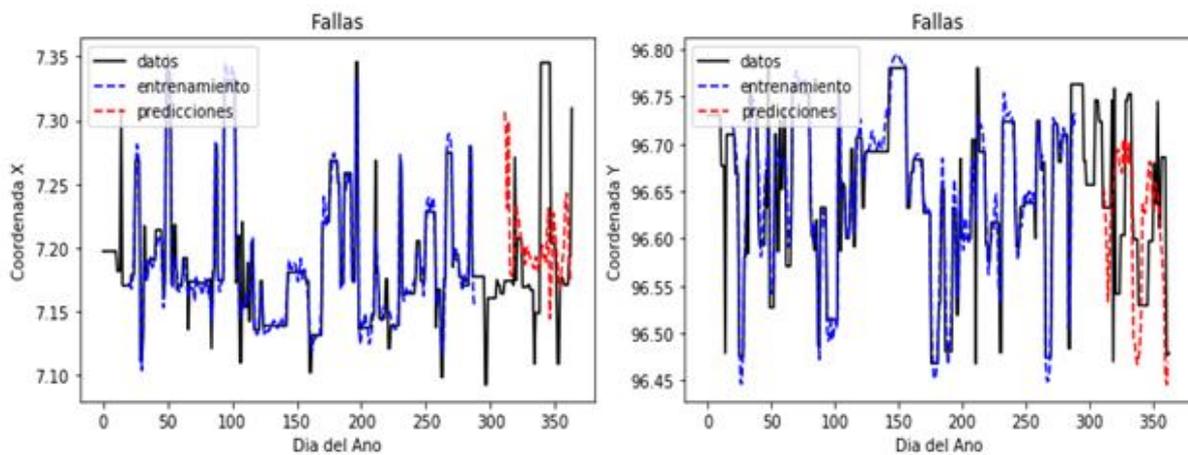
**Figura 4.11** Predicciones de coordenadas X e Y, red sin capas ocultas, *batch* = 2



**Figura 4.12** Predicciones de coordenadas X e Y, red sin capas ocultas, *batch* = 5



**Figura 4.13** Predicciones de coordenadas X e Y, red sin capas ocultas, *batch* = 10



**Figura 4.14** Predicciones de coordenadas X e Y, red sin capas ocultas, *batch* = 15

#### 4.4.1.2 Modelo de una capa oculta:

##### 4.4.1.2.1 Variación del hiper parámetro *batch* con una capa oculta

En las figuras 4.15 - 4.18 se muestran los resultados de la predicción en una red con una capa oculta en base a la variación del *batch*. Las Tablas 4.3 y 4.4 muestran el error correspondiente a cada una de estas variaciones, concluyendo que el error mínimo se da con *batch* = 2, concluyendo en un resultado satisfactorio según lo esperado.

**Tabla 4.3** Error de una red con una capa oculta para diferentes *batch*, predicción de coordenadas X.

| Coordenada X      | RMSE Entrenamiento | RMSE Pruebas |
|-------------------|--------------------|--------------|
| Capas Ocultas     | 1                  | 1            |
| <i>Batch</i> = 2  | 0.01               | 0.06         |
| <i>Batch</i> = 5  | 0                  | 0.06         |
| <i>Batch</i> = 10 | 0.01               | 0.07         |
| <i>Batch</i> = 15 | 0                  | 0.07         |

**Tabla 4.4** Error de una red con una capa oculta para diferentes *batch*, predicción de coordenadas Y.

| Coordenada Y      | RMSE Entrenamiento | RMSE Pruebas |
|-------------------|--------------------|--------------|
| Capas Ocultas     | 1                  | 1            |
| <i>Batch</i> = 2  | 0.01               | 0.08         |
| <i>Batch</i> = 5  | 0.01               | 0.08         |
| <i>Batch</i> = 10 | 0.01               | 0.09         |
| <i>Batch</i> = 15 | 0.01               | 0.09         |

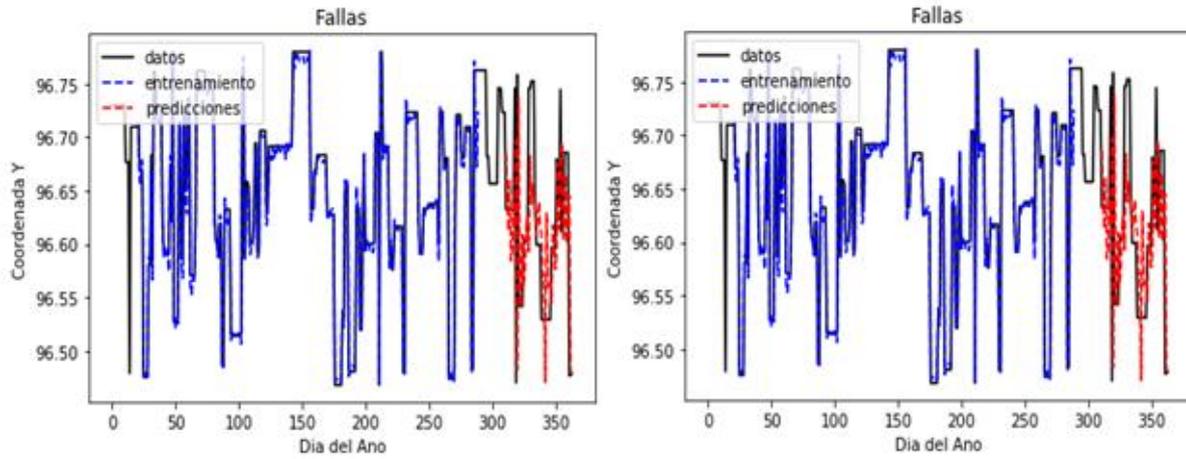


Figura 4.15 Predicciones de coordenadas X e Y, red de una capa oculta,  $batch = 2$

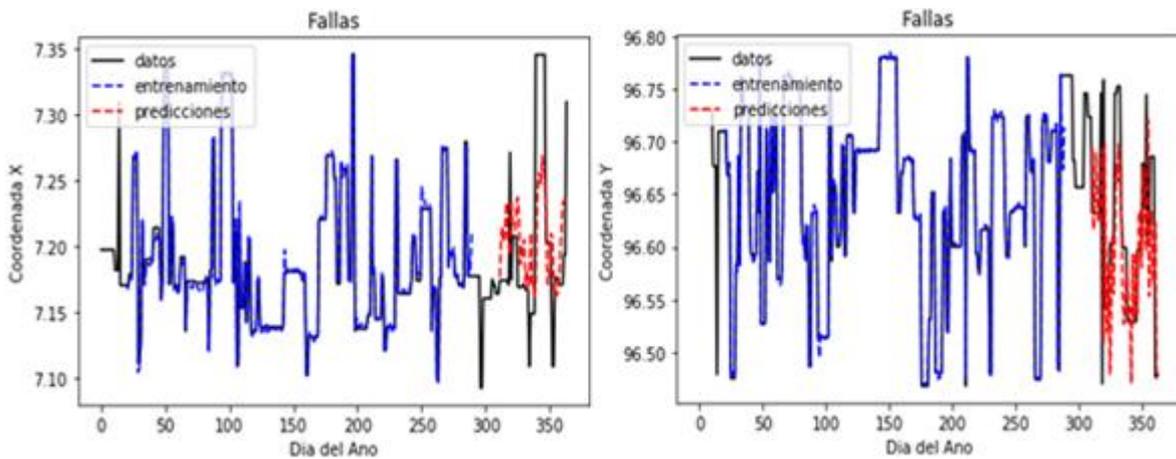


Figura 4.16 Predicciones de coordenadas X e Y, red de una capa oculta,  $batch = 5$

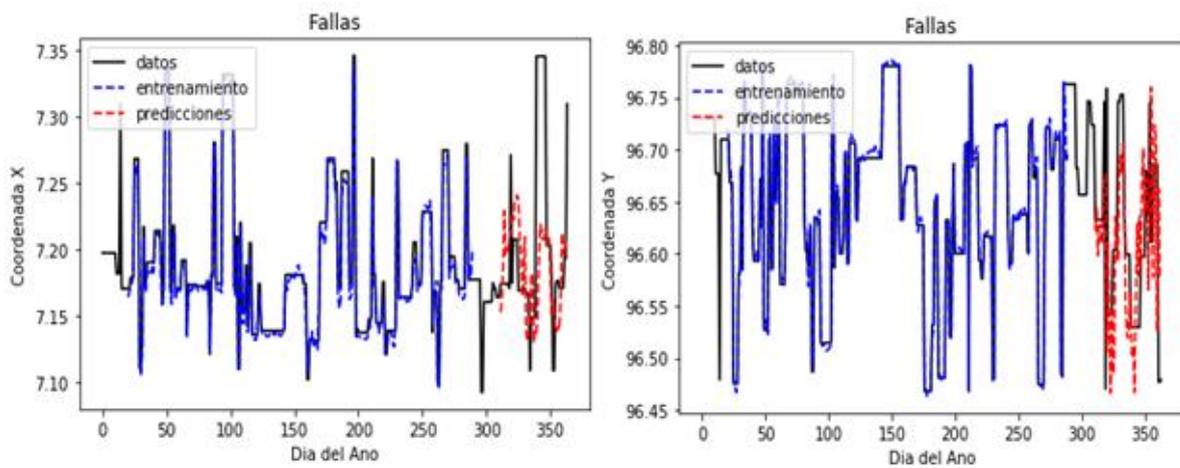


Figura 4.17 Predicciones de coordenadas X e Y, red de una capa oculta,  $batch = 10$

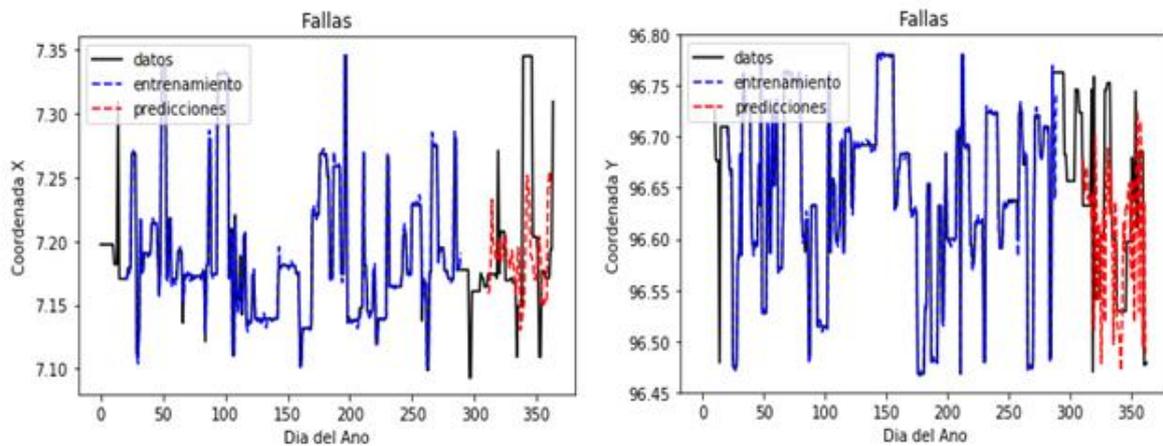


Figura 4.18 Predicciones de coordenadas X e Y, red de una capa oculta,  $batch = 15$

#### 4.4.1.3 Modelo de dos capas ocultas

##### 4.4.1.3.1 Variación del hiper parámetro $batch$ con dos capas ocultas

En las figuras, desde la 4.19 hasta la 4.22, se muestran los resultados de la predicción en una red con dos capas ocultas en base a la variación del  $batch$ . Las tablas 4.5 y 4.6 muestran el error correspondiente a cada una de estas variaciones, concluyendo que el error mínimo se da con  $batch = 2$ , siendo esto lo esperado según lo revisado anteriormente.

Tabla 4.5 Error de una red con dos capas ocultas para diferentes  $batch$ , predicción de coordenadas X.

| Coordenada X         | RMSE Entrenamiento | RMSE Pruebas |
|----------------------|--------------------|--------------|
| <b>Capas Ocultas</b> | <b>2</b>           | <b>2</b>     |
| <i>Batch = 2</i>     | 0                  | 0.06         |
| <i>Batch = 5</i>     | 0                  | 0.07         |
| <i>Batch = 10</i>    | 0.01               | 0.07         |
| <i>Batch = 15</i>    | 0.01               | 0.08         |

Tabla 4.6 Error de una red con dos capas ocultas para diferentes  $batch$ , predicción de coordenadas Y.

| Coordenada Y         | RMSE Entrenamiento | RMSE Pruebas |
|----------------------|--------------------|--------------|
| <b>Capas Ocultas</b> | <b>2</b>           | <b>2</b>     |
| <i>Batch = 2</i>     | 0.01               | 0.08         |
| <i>Batch = 5</i>     | 0                  | 0.08         |
| <i>Batch = 10</i>    | 0.01               | 0.08         |
| <i>Batch = 15</i>    | 0.01               | 0.09         |

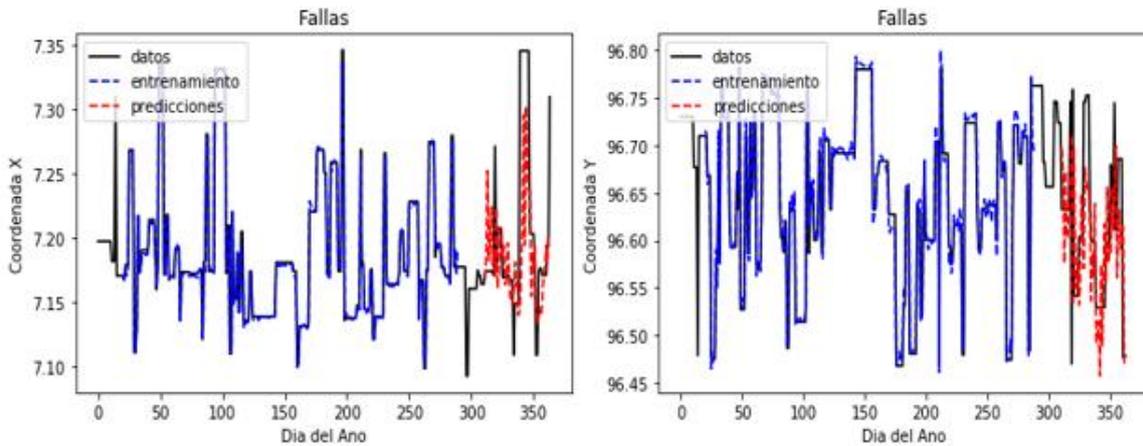


Figura 4.19 Predicciones de coordenadas X e Y, red de dos capas ocultas, *batch = 2*

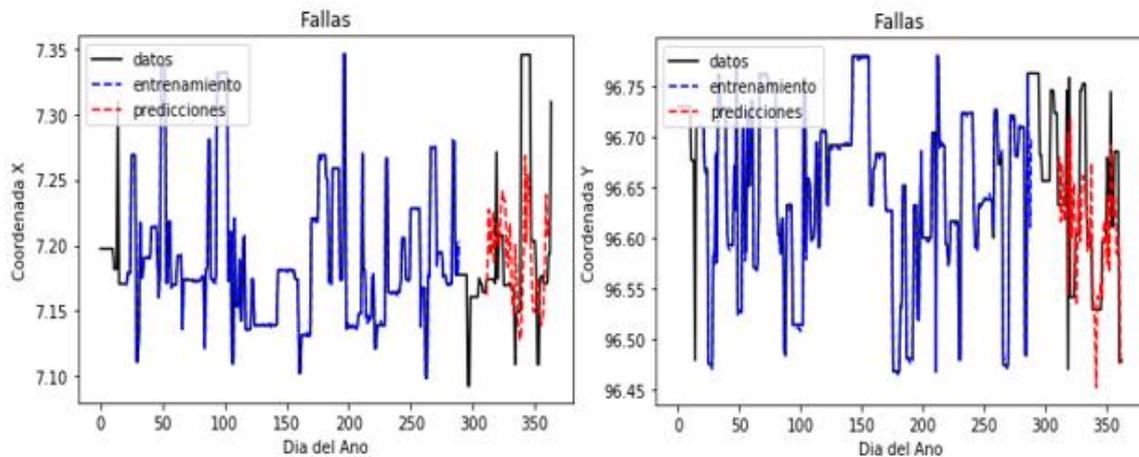


Figura 4.20 Predicciones de coordenadas X e Y, red de dos capas ocultas, *batch = 5*

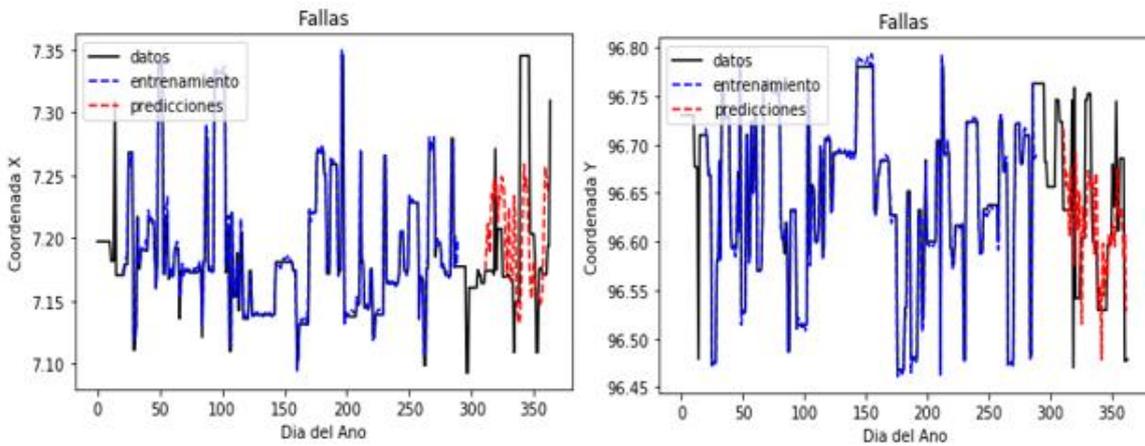


Figura 4.21 Predicciones de coordenadas X e Y, red de dos capas ocultas,  $batch = 10$

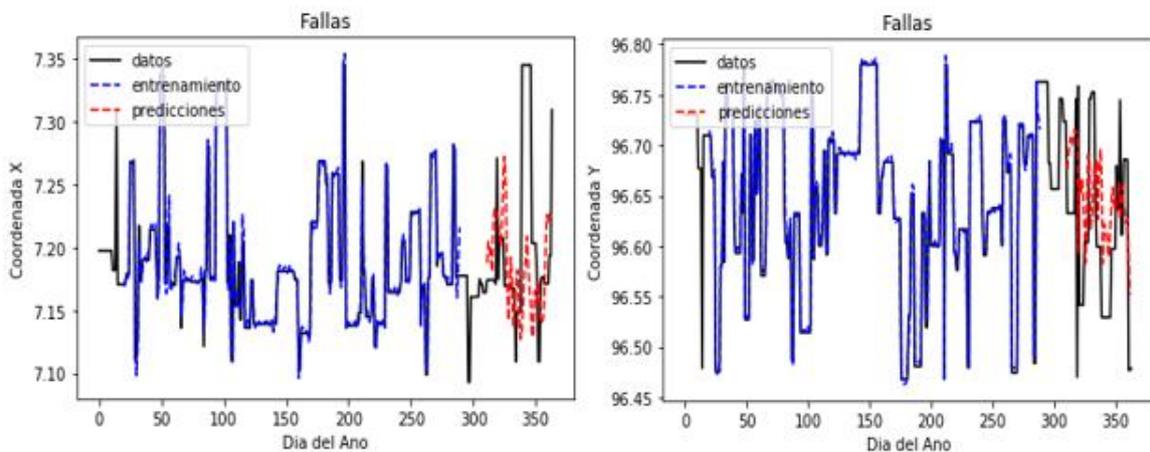


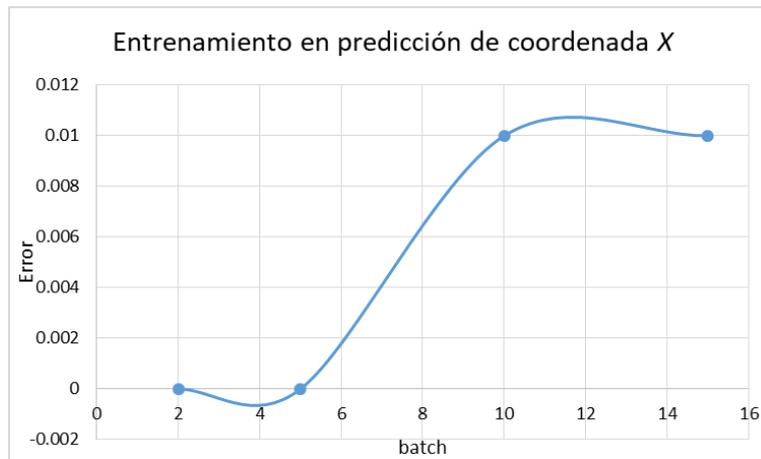
Figura 4.22 Predicciones de coordenadas X e Y, red de dos capas ocultas,  $batch = 15$

#### 4.4.1.4 El error versus el aprendizaje por lotes

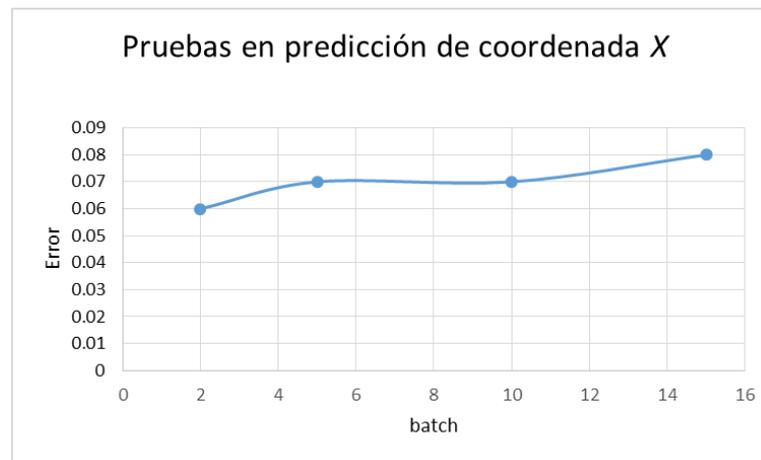
Se demostró que a mayor cantidad de capas ocultas y número  $batch$  más pequeño se obtienen errores menores (evaluando las predicciones X e Y) como se observa en las Tablas 4.7 y 4.8, por lo que para las siguientes pruebas esos dos parámetros quedan fijos.

**Tabla 4.7** Resultados del error en base al número de capas ocultas de la red y el *batch* para la predicción de coordenadas X

| Coordenada X      | RMSE Entrenamiento |      |      | RMSE Pruebas |      |      |
|-------------------|--------------------|------|------|--------------|------|------|
|                   | 0                  | 1    | 2    | 0            | 1    | 2    |
| <i>Batch</i> = 2  | 0.01               | 0.01 | 0    | 0.07         | 0.06 | 0.06 |
| <i>Batch</i> = 5  | 0.01               | 0    | 0    | 0.07         | 0.06 | 0.07 |
| <i>Batch</i> = 10 | 0.02               | 0.01 | 0.01 | 0.08         | 0.07 | 0.07 |
| <i>Batch</i> = 15 | 0.01               | 0    | 0.01 | 0.07         | 0.07 | 0.08 |



**Figura 4.23** RMSE vs *batch* en el entrenamiento de dos capas ocultas para X



**Figura 4.24** RMSE vs *batch* en las pruebas de dos capas ocultas para X

**Tabla 4.8** Resultados del error en base al número de capas ocultas de la red y el *batch* para la predicción de coordenadas Y

| Coordenada Y      | RMSE Entrenamiento |      |      | RMSE Pruebas |      |      |
|-------------------|--------------------|------|------|--------------|------|------|
| Capas Ocultas     | 0                  | 1    | 2    | 0            | 1    | 2    |
| <i>Batch</i> = 2  | 0.02               | 0.01 | 0.01 | 0.1          | 0.08 | 0.08 |
| <i>Batch</i> = 5  | 0.01               | 0.01 | 0    | 0.09         | 0.08 | 0.08 |
| <i>Batch</i> = 10 | 0.03               | 0.01 | 0.01 | 0.11         | 0.09 | 0.08 |
| <i>Batch</i> = 15 | 0.03               | 0.01 | 0.01 | 0.09         | 0.09 | 0.09 |

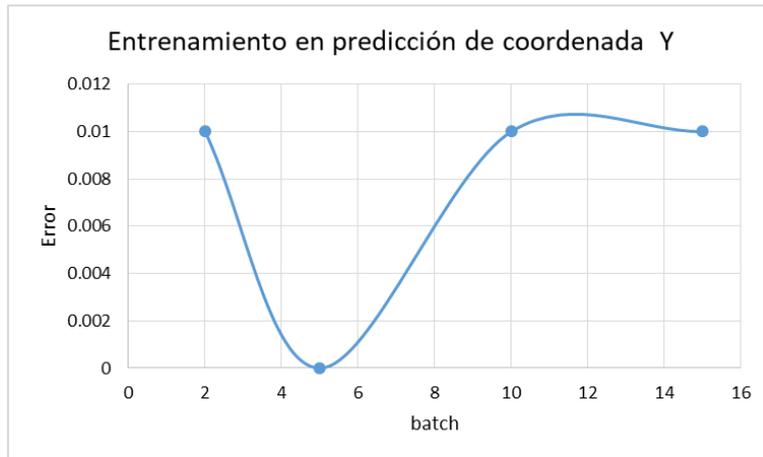


Figura 4.25 RMSE vs *batch* en el entrenamiento de dos capas ocultas para Y

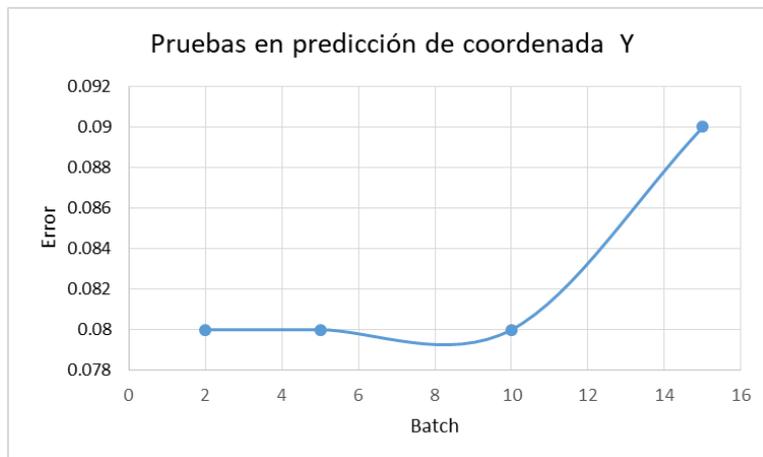


Figura 4.26 RMSE vs *batch* en las pruebas de dos capas ocultas para Y

#### 4.4.1.5 Funciones de activación en la capa de salida

Como se observa en las Tabla 4.9 y Tabla 4.10 las funciones con mayor tendencia lineal presentan cambios nulos en el error y la función *tanh*, por su híper-



simetría con los ejes cartesianos, induce errores mayores en la salida. Por lo que se concluye que en la capa de salida, una función lineal es la mejor opción.

**Tabla 4.9** Error para diferentes funciones de activación en la capa de salida de un modelo de dos capas ocultas de la predicción de coordenada X

| Coordenada X       | <i>Linear</i> | <i>Relu</i> | <i>tanh</i> |
|--------------------|---------------|-------------|-------------|
| RMSE Entrenamiento | 0             | 0           | 0.01        |
| RMSE Pruebas       | 0.06          | 0.06        | 0.8         |

**Tabla 4.10** Error para diferentes funciones de activación en la capa de salida de un modelo de dos capas ocultas de la predicción de coordenada Y

| Coordenada Y       | <i>Linear</i> | <i>Relu</i> | <i>tanh</i> |
|--------------------|---------------|-------------|-------------|
| RMSE Entrenamiento | 0.01          | 0.01        | 0.02        |
| RMSE Pruebas       | 0.08          | 0.08        | 0.09        |

## 4.5 PREDICCIONES NULAS

### 4.5.1.1 Prueba con parámetros aleatorios

El objetivo de esta prueba es mostrar predicciones totalmente erróneas que se manifiestan cuando no se ha planteado algún criterio de selección de hiperparámetros en una red neuronal, un modelo fue realizado con el siguiente ajuste:

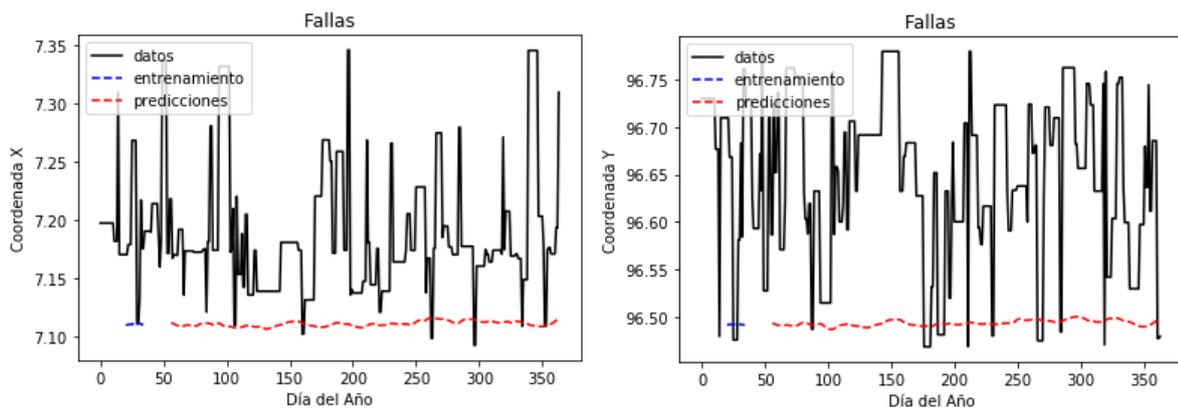
- i. Iteraciones o *epoch* = 1 (contrastando que una iteración es un número pequeño para cualquier proceso repetitivo complejo como este problema)
- ii. Porcentaje de datos para el entrenamiento = 10% (contrastando que la literatura recomienda de 65% a 85%)

- iii. *Batch* = 365 (contrastando que las series de tiempo deberían usar un *batch* pequeño)
- iv. Función de activación sigmoide en la capa de salida (contrastando el poco uso dado en los modelos de APM)

En la Tabla 4.11 se observan errores de magnitud considerable, por lo que se espera malas predicciones de parte de este modelo, de la misma forma en la Figura 4.27 se puede apreciar el fallido intento del algoritmo por realizar la predicción.

**Tabla 4.11** Error para una red configurada sin criterio de selección de parámetros

| Predicciones | Rmse Entrenamiento | RMSE Pruebas |
|--------------|--------------------|--------------|
| Coordenada X | 0.29               | 0.18         |
| Coordenada y | 0.19               | 0.22         |



**Figura 4.27** Predicción de coordenadas X e Y con parámetros aleatorios

#### 4.6 RESULTADOS EN ORDENADORES CON DIFERENTE PLATAFORMA DE HARDWARE Y SOFTWARE

El APM presenta propiedades estocásticas, por esta razón, con la finalidad de diseñar un método confiable y aplicable en cualquier ordenador, se usaron 2 computadores de características diferentes para probar el modelo, cuyas características se muestran en la Tabla 4.12.

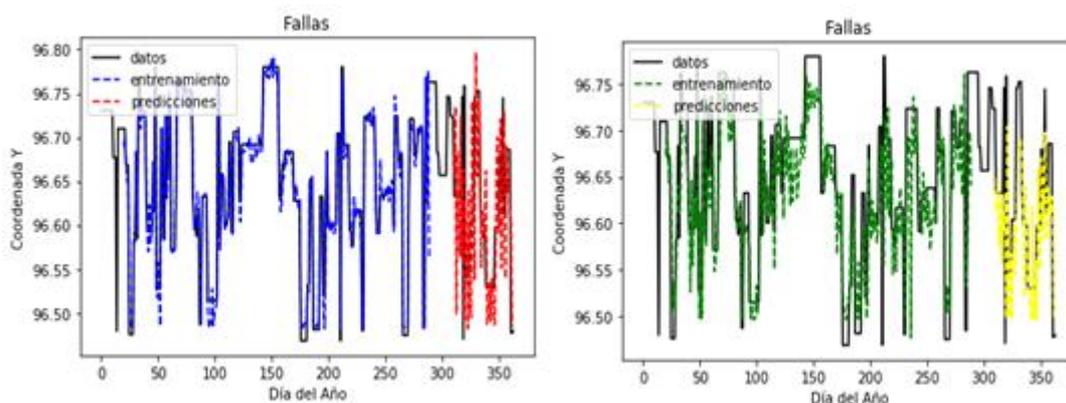
**Tabla 4.12** Características de los diferentes ordenadores

| Ordenador       | 1                | 2                        |
|-----------------|------------------|--------------------------|
| Firma           | HEWLETT PACKARD  | ASUS                     |
| Procesador      | INTEL I7//2.5GHz | INTEL I7 7Th gen//2.8GHz |
| Tarjeta Gráfica | Estandar         | Nvidia Gforce GTX        |
| RAM             | 8Gb              | 8Gb                      |
| ROM             | 512Gb            | 1Tb                      |
| SO              | Windows 10       | Ubuntu(basado en Linux)  |

El modelo se probó en la versión de CPU (Ordenador 1) de Tensorflow así como también en la versión de GPU (Ordenador 2), mostrando el Ordenador 2 un poco menos de tiempo en completar los *epochs* con un error ligeramente más pequeño como se detalla en la Tabla 4.13, también en la Figura 4.28 se puede ver esta leve mejoría en cuanto a predicción. Se concluye que la diferencia no es muy amplia.

**Tabla 4.13** Errores en la predicción de coordenadas Y en 2 ordenadores

| Coordenada Y | Rmse Entrenamiento | RMSE Pruebas |
|--------------|--------------------|--------------|
| GPU          | 0.01               | 0            |
| CPU          | 0.03               | 0.03         |



**Figura 4.28** Predicción de la coordenada Y (izquierda: GPU, derecha: CPU)



## CAPÍTULO 5

### 5 CONCLUSIONES Y RECOMENDACIONES

En este trabajo de titulación se desarrolló un modelo predictivo de fallas en alimentadores primarios de concesión de la CENTROSUR.

Para problemas de predicción relacionados a un evento en el tiempo, los modelos de APM son muy efectivos. Existen hoy en día herramientas que facilitan la construcción de las redes neuronales, como la librería Keras que corre sobre TensorFlow. Estas herramientas hacen que en pocas líneas de código se pueda construir cualquier arquitectura compleja de red neuronal de forma sencilla, rápida y precisa.

El desarrollo del modelo computacional para la predicción de fallas usando APM en un alimentador primario de la CENTROSUR resulta de interés teórico debido a que:

- i. Las series de tiempo pueden modelar muchos fenómenos relacionados a un sistema de potencia, en este caso particular a la ocurrencia de fallas en un alimentador de distribución.
- ii. Este trabajo comprobó que no es óptimo incluir una cantidad exuberante de iteraciones para conseguir resultados positivos en un proceso iterativo.
- iii. Se comprobó que el aprendizaje por lotes (*batch*) resulta mejor si se lo considera como un valor pequeño dentro de los hiper-parámetros de una red neuronal que persiga como fin predicciones en series de tiempo.
- iv. Fue apreciado el resultado positivo de incluir una o dos capas de neuronas ocultas en una red, y se constató que una función Relu aparte de ser muy popular estos días, es de beneficio en la salida de una RMLP.



El desarrollo del modelo computacional para la predicción de fallas usando APM en un alimentador primario de la CENTROSUR resulta de interés práctico debido a que:

- i. El APM no ha sido explorado por la industria eléctrica en problemas similares, por lo cual el uso de APM para resolver el problema de predicción de fallas resulta de interés de las compañías de distribución eléctrica locales e internacionales para aplicaciones pioneras
- ii. El APM puede ser usado virtualmente en todas las áreas en las que trabaje una empresa, pueden verse beneficiados desde la economía hasta la ingeniería.
- iii. La CENTROSUR presenta un registro de fallas, algunas repetitivas, todos los meses del año y actualmente no cuenta con ninguna herramienta tecnológica que sea capaz de predecir la ocurrencia de fallas. Esta herramienta podría aportar mucho a la empresa de manera económica
- iv. Los sistemas para la recopilación de información como el SCADA, toman mayor protagonismo a la hora de analizar grandes cantidades de datos. Por esto, la utilidad de este trabajo resulta relevante si esos datos pueden servir para la creación de algoritmos de predicción

Los resultados de esta investigación muestran que es posible desarrollar estos sofisticados modelos prácticamente en cualquier empresa donde se cuente con información recolectada. Esto puede contribuir al desarrollo de *Big Data*, tan importante en esta década, sin contar que hay muchas herramientas de Software libre a plena disposición para su investigación.

Se puede recomendar a la CENTROSUR que, aprovechando su enorme cantidad de datos guardados a través de los últimos años en, haga uso pleno de



ellos para entrenar modelos predictores de diferentes fenómenos. Es recomendable también, la obtención o desarrollo de sensores para obtener más características de una falla, como, por ejemplo; temperatura, velocidad del viento, precipitación, radiación solar y otras características que pueden ser registradas durante una interrupción. Finalmente se recomienda que la empresa haga énfasis en el registro del último mantenimiento en cada alimentador después de que sufre una falla, esta es una de las características más importantes que podría haber aportado mucho a los resultados obtenidos en este trabajo de titulación.



## BIBLIOGRAFÍA

- [1] Radmer, D., Kuntz, P., Christie, R., Venkata, S. and Fletcher, R. (2002). "Predicting Vegetation Related Failure Rates for Overhead Distribution Feeders". IEEE Power Engineering Review, 22(9), pp.64-64.
- [2] Santa, B. "Estudios de Predicción en Series Temporales de Datos Meteorológicos Utilizando Redes Neuronales Recurrentes". Título para Ingeniero Informático. Universidad de las Palmas de Gran Canari. (2016). Pág. 1-50.
- [3] Mora, M. "Fundamentos de Deep Learning y una Aplicación para Predecir la Demanda de Energía Eléctrica en México". Licenciado en Matemáticas Aplicadas. Instituto Tecnológico Autónomo de México. (2016). Pág. 1-137.
- [4] Cynthia Rudin, Seyda Ertekin, Rebecca Passonneau, Axinia Radeva, Ashish Tomar, Boyi Xie, Stanley Lewis, Mark Riddle, Debbie Pangsrivini, John Shipman, Tyler McCormick. "Analytics for Power Grid Distribution Reliability in New York City". Información tecnológica consolidada por la empresa Con Edison Con Edison. (2014). Pág. 1-33.
- [5] Philip Gross, Albert Boulanger, Marta Arias, David Waltz, Philip M. Long, Charles Lawson, Roger Anderson Matthew Koenig, Mark Mastrocinque, William Fairechio, John A. Johnson, Serena Lee, Frank Doherty, Arthur Krasner. "Predicting Electricity Distribution Feeder Failures Using Machine Learning Susceptibility Analysis". Información tecnológica consolidada por la empresa Con Edison. Columbia University. (2005). Pág. 1-7.
- [6] Análisis y Diagnóstico de Sistemas Eléctricos de Potencia – LIELECTRIC, CA. Accedido el 10 de septiembre, 2017 desde: <http://www.lielectric.com.ve/analisis-y-diagnostico-de-sistemas-electricos-de-potencia/>.



- [7] Seymour, J., & Horsley, T. Informe interno N° 18. Estados Unidos. American Power Conversion. “Los siete tipos de problemas en el suministro eléctrico”. (2005). Pág. 6-23.
- [8] “Normalización del mantenimiento preventivo de líneas aéreas de distribución de energía eléctrica de tensión entre 6 KV y 72,5 KV”. Documento técnico. Montevideo-Uruguay. UTE. (2006). Pág. 1-19.
- [9] Crespo, E. “Desarrollo de un modelo para la localización de fallas en sistemas de transmisión de energía eléctrica utilizando técnicas de inteligencia artificial”. Maestría en ingeniería en sistemas. Universidad Tecnológica de Medellín. (2010). Pág.8-13.
- [10] Área de Concesión | Centrosur. Accedido el 1 de septiembre, 2017 desde <http://www.centrosur.gob.ec/?q=node/11>.
- [11] Zaruma, J., & Blacio, D. “Análisis de confiabilidad del sistema de distribución de la Empresa Eléctrica Regional Centrosur C. A”. Posgrado en ingeniería eléctrica. Universidad de Cuenca. (2012). Pág. 13-38.
- [12] CONELEC, “Calidad del servicio eléctrico de distribución”. *Regulación No. CONELEC 004/01*. (2001)
- [13] D, Balbotín Noval, F.J Martin Mateos, J.L Ruiz Reina. “Introducción a las redes neuronales”. D, Balbotín Noval, F.J Martin Mateos, J.L Ruiz Reina. Departamento ciencias de la computación e inteligencia artificial. Universidad de Sevilla (2013).
- [14] Pedro Larrañaga, Iñaki Inza, Abdelmalik Moujahid. “Redes Neuronales”. Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad del País Vasco. (2017).



[15] Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs – WildML. Accedido el 24 de octubre, 2017 desde: <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>

[16] Understanding LSTM Networks – colah’s blog. Accedido el 8 de octubre, 2017 desde: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

[17] Tom Mitchell's Home Page. Accedido el 21 de octubre, del 2017 desde: Desde <http://www.cs.cmu.edu/~tom/>.

[18] El deep learning está de moda: así hacen Google, Apple y Facebook que los ordenadores piensen como humanos – El diario.es. Accedido el 20 de octubre, 2017 desde: [http://www.eldiario.es/hojaderouter/tecnologia/software/moda-deep\\_learning-algoritmo-inteligencia\\_artificial\\_0\\_275772610.html](http://www.eldiario.es/hojaderouter/tecnologia/software/moda-deep_learning-algoritmo-inteligencia_artificial_0_275772610.html).

[19] Deep Learning’s Next Frontier – Harvard Business Review. (2017). Accedido el 8 de octubre, del 2017 desde: <https://hbr.org/webinar/2017/07/deep-learnings-next-frontier>.

[20] “Big Data Tsunami at the interface of statics, environmental sciences and beyond” Yulia R, Lyubchich V, Ramirez L. Centro de investigación. (2016). Pág. 1-2.

[21] World Bank. Accedido el 24 de septiembre, del 2017 desde: <http://www.bancomundial.org.^>

[22] “Las coordenadas geográficas y la proyección UTM”. Fernández I. Área de Ingeniería Cartográfica. Universidad de Valladolid. (2001). Pág.1-86.

[23] Igm. Gob. ec. Accedido el 25 de septiembre, del 2017 desde: [http://www.igm.gob.ec/work/files/cartabase/n/NV\\_F4.htm](http://www.igm.gob.ec/work/files/cartabase/n/NV_F4.htm).



[24] Docs. Microsoft.com. Conjunto de datos de entrenamiento y de prueba. Accedido el 25 de septiembre, del 2017 desde: <https://docs.microsoft.com/es-es/sql/analysis-services/data-mining/training-and-testing-data-sets>.

[25] CONELEC. “Plan maestro de electrificación 2012-2021”. (2012).

[26] ARCONELC. “Ley Orgánica del Servicio Público de Energía Eléctrica”. (2015).

[27] Gómez , Víctor A; . Peña, Robin A y Hernández, Cesar. “Identificación y Localización de Fallas en Sistemas de Distribución con Medidores de Calidad del Servicio de Energía Eléctrica”. Información Tecnológica. (2012). Pág. 109-106.

[28] Bedoya, J. “Algoritmos para localización de fallas en sistemas de distribución usando máquinas de soporte vectorial”. Maestría. Universidad Tecnológica de Pereira. (2010). Pág. 1-6.

[29] Kandallu, R., Sah, P. and Agrawal, S. “Harnessing the Power of IoT in Manufacturing: Predicting Parts Failure with the Right Analytical Model”. Tata Consultancy services. Documento técnico. Tata consultancy service. (2015). Pág. 1-13.

[30] Faisal, M. and Mohamed, A. “Support Vector Regression Based S-transform for Prediction of Distribution Network failure”. Trabajo de investigación. Universidad Kebangsaan. Malasia. (2006). Pág. 1-6.

[31] Pérez C, Santín D. “Minería de datos técnicas y herramientas”. Documento. Madrid-España. Paraninfo, S.A. (2008).

[32] Provost F, Fawcett T. “Data Science for Business”. Documento. United States of America. Copyright. (2013).

[33] Sedgewick R, Wayne K. “Algorithms”. Trabajo de investigación. Princeton University. United States. Copyright. (2011).



# ANEXOS



## ANEXOS

### ANEXO A

Tablas que registran el número interrupciones no programadas que se han generado en los años 2011, 2012, 2013, 2014 Y 2015

#### AÑO 2011

**Tabla A.1** Total, de interrupciones registradas para el año 2011

| DESCRIPCIÓN                          | CAUSA      | TOTAL       |
|--------------------------------------|------------|-------------|
| Cortocircuito                        | Red        | 814         |
| Cambio de fusible                    | Red        | 664         |
| Ramas sobre la red                   | Terceros   | 621         |
| Descargas atmosféricas               | Climáticas | 412         |
| Línea arrancada                      | Red        | 273         |
| Viento                               | Climáticas | 255         |
| Red floja                            | Red        | 189         |
| Cambiar bases de portafusibles en BT | Red        | 148         |
| Choque                               | Terceros   | 86          |
| Cambio de conector                   | Red        | 71          |
| Falla transitoria                    | Red        | 70          |
| Cambiador tirafusibles               | Red        | 54          |
| Cambiar poste                        | Red        | 36          |
| Otras                                | Red        | 29          |
| Tempestad o tormenta                 | Climáticas | 29          |
| Actúa Relé de sobrecorriente         | Red        | 29          |
| Falla del equipo de medición         | Red        | 15          |
| Animales en la red                   | Terceros   | 14          |
| Tensor suelto                        | Red        | 10          |
| <b>Total</b>                         |            | <b>3819</b> |



AÑO 2012

Tabla A.2 Total, de interrupciones registradas para el año 2012

| DESCRIPCIÓN                       | CAUSA       | TOTAL       |
|-----------------------------------|-------------|-------------|
| Cortocircuito                     | Red         | 845         |
| Ramas sobre la red                | Terceros    | 662         |
| Descargas atmosféricas            | Climáticas  | 456         |
| Cambio de fusibles                | Red         | 393         |
| Red floja                         | Red         | 271         |
| Línea arrancada                   | Red         | 255         |
| Viento                            | Climáticas  | 192         |
| Cambiar bases de portafusibles BT | Red         | 189         |
| Otras                             | Red         | 124         |
| Choque                            | Terceros    | 118         |
| Deslizamiento de tierra           | Ambientales | 67          |
| Tempestad o tormenta              | Climáticas  | 15          |
| Actúa réle de sobrecorriente      | Red         | 70          |
| Actúa sistemas de protecciones    | Red         | 14          |
| Cambiar tirafusibles              | Red         | 33          |
| Falla transitoria                 | Red         | 35          |
| Trafo o red sobrecargada          | Red         | 20          |
| Animales en la red                | Terceros    | 21          |
| <b>Total</b>                      |             | <b>3780</b> |



AÑO 2013

**Tabla A.3** Total, de interrupciones registradas para el año 2013

| DESCRIPCIÓN                    | CAUSA      | TOTAL       |
|--------------------------------|------------|-------------|
| Cortocircuito                  | Red        | 593         |
| Ramas sobre la red             | Terceros   | 588         |
| Descargas atómicas             | Climáticas | 432         |
| Otras                          | Red        | 344         |
| Cambio de fusibles             | Red        | 317         |
| Cambiar fases de portafusibles | Red        | 205         |
| Red floja                      | Red        | 171         |
| Viento                         | Climáticas | 138         |
| Línea arrancada                | Red        | 127         |
| Choque                         | Terceros   | 126         |
| Animales en la red             | Terceros   | 47          |
| Cambiar tirafusibles           | Red        | 22          |
| Falla transitoria              | Red        | 20          |
| Actúa rele de sobrecorriente   | Red        | 18          |
| Cambio de conector             | Red        | 16          |
| Tempestad o tormenta           | Red        | 13          |
| Daño interno                   | Terceros   | 13          |
| Red sobrecargada               | Red        | 11          |
| Transformador quemado          | Red        | 10          |
| Cambio de poste                | Red        | 8           |
| <b>Total</b>                   |            | <b>3219</b> |



AÑO 2014

**Tabla A.4** Total, de interrupciones registradas para el año 2014

| DESCRIPCIÓN                    | CAUSA      | TOTAL       |
|--------------------------------|------------|-------------|
| Cortocircuito                  | Red        | 910         |
| Ramas sobre la red             | Terceros   | 650         |
| Descargas atmosféricas         | Climáticas | 515         |
| Otras                          | Red        | 324         |
| Cambio de fusibles             | Red        | 261         |
| Cambiar bases de portafusibles | Red        | 204         |
| Línea arrancada                | Red        | 182         |
| Choque                         | Terceros   | 163         |
| Red floja                      | Red        | 157         |
| Cambiar tirafusibles           | Red        | 72          |
| Viento                         | Climáticas | 69          |
| Animales en la red             | Terceros   | 50          |
| Tempestad o tormenta           | Climáticas | 45          |
| Actúa relé de sobrecorriente   | Red        | 41          |
| Daño interno                   | Terceros   | 16          |
| Falla transitoria              | Red        | 13          |
| Cambio de conector             | Red        | 12          |
| Trafo o red sobrecargada       | Red        | 9           |
| Cambiar poste                  | Red        | 8           |
| <b>Total</b>                   |            | <b>3701</b> |



AÑO 2015

**Tabla A.5** Total, de interrupciones registradas para el año 2015

| DESCRIPCIÓN                        | CAUSA      | TOTAL       |
|------------------------------------|------------|-------------|
| Cortocircuito                      | Red        | 1059        |
| Ramas sobre la red                 | Terceros   | 726         |
| Descargas atmosféricas             | Climáticas | 350         |
| Cambiar bases del portafusibles BT | Red        | 307         |
| Línea arrancada                    | Red        | 220         |
| Falla transitoria                  | Red        | 196         |
| Otras                              | Red        | 167         |
| Viento                             | Climáticas | 166         |
| Choque                             | Terceros   | 141         |
| Red floja                          | Red        | 140         |
| Cambio de fusibles                 | Red        | 77          |
| Cambio de conector                 | Red        | 38          |
| Cambiar tirafusibles               | Red        | 37          |
| Actúa sistema de protecciones      | Red        | 31          |
| Puente en MT arrancado             | Red        | 26          |
| Animales en la red                 | Terceros   | 23          |
| Actúa relé de sobrecorriente       | Red        | 23          |
| Seccionador averiado               | Red        | 15          |
| Transformador quemado              | Red        | 12          |
| Cambiar poste                      | Red        | 12          |
| <b>Total</b>                       |            | <b>3766</b> |



## **ANEXO B**

### **DATOS ESTADÍSTICOS DE LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR C.A.**

La empresa eléctrica proporcionó información relevante con respecto a las interrupciones no programadas que presentaron los años 2011, 2012, 2013, 2014, 2015 y 2016; la siguiente tabla es solo una muestra del tipo de información que se dispuso para este trabajo, aunque no representa la totalidad de la información, tiene como objetivo dar una idea de la información recopilada a lo largo de los últimos años.

El total de la información recopilada por la empresa y usada para la elaboración del método es del alimentador 521 en cinco años.

**Tabla B.1** Muestra parcial de la información de fallas y sus características del alimentador 521 desde enero del 2011 hasta diciembre del 2015

| Anio | MesVisualizacion | Alim   | Fecha Desconexion | Causa      | CoordenadaX | CoordenadaY  |
|------|------------------|--------|-------------------|------------|-------------|--------------|
| 2011 | ENERO            | 521.00 | 1/1/2011          | Red        | 717,758.21  | 9,676,478.38 |
| 2011 | FEBRERO          | 521.00 | 2/12/2011         | Red        | 718,935.20  | 9,673,088.23 |
| 2011 | FEBRERO          | 521.00 | 2/17/2011         | Climaticas | 717,957.05  | 9,678,145.69 |
| 2011 | FEBRERO          | 521.00 | 2/19/2011         | Red        | 713,761.39  | 9,671,224.73 |
| 2011 | FEBRERO          | 521.00 | 2/20/2011         | Red        | 716,998.21  | 9,673,630.93 |
| 2011 | MARZO            | 521.00 | 3/11/2011         | Red        | 716,810.08  | 9,675,263.72 |
| 2011 | MARZO            | 521.00 | 3/19/2011         | Red        | 717,145.89  | 9,673,800.38 |
| 2011 | MARZO            | 521.00 | 3/21/2011         | Climaticas | 726,817.10  | 9,647,520.86 |
| 2011 | MARZO            | 521.00 | 3/30/2011         | Red        | 716,915.10  | 9,672,660.44 |
| 2011 | ABRIL            | 521.00 | 4/1/2011          | Terceros   | 716,998.21  | 9,673,630.93 |
| 2011 | ABRIL            | 521.00 | 4/1/2011          | Terceros   | 716,440.65  | 9,664,050.51 |
| 2011 | ABRIL            | 521.00 | 4/1/2011          | Climaticas | 716,330.68  | 9,664,024.29 |
| 2011 | ABRIL            | 521.00 | 4/5/2011          | Climaticas | 722,199.62  | 9,653,674.87 |
| 2011 | ABRIL            | 521.00 | 4/13/2011         | Red        | 719,095.13  | 9,667,129.73 |
| 2011 | ABRIL            | 521.00 | 4/17/2011         | Red        | 717,145.89  | 9,673,800.38 |
| 2011 | ABRIL            | 521.00 | 4/19/2011         | Red        | 717,793.87  | 9,677,244.47 |
| 2011 | ABRIL            | 521.00 | 4/19/2011         | Red        | 713,598.24  | 9,659,985.67 |
| 2011 | ABRIL            | 521.00 | 4/19/2011         | Red        | 717,095.53  | 9,659,937.69 |
| 2011 | ABRIL            | 521.00 | 4/20/2011         | Red        | 717,559.63  | 9,659,965.03 |
| 2011 | ABRIL            | 521.00 | 4/21/2011         | Terceros   | 719,944.30  | 9,667,962.90 |
| 2011 | ABRIL            | 521.00 | 4/21/2011         | Red        | 719,944.30  | 9,667,962.90 |
| 2011 | ABRIL            | 521.00 | 4/25/2011         | Red        | 717,978.42  | 9,670,874.77 |
| 2011 | ABRIL            | 521.00 | 4/30/2011         | Terceros   | 714,896.97  | 9,670,431.07 |
| 2011 | ABRIL            | 521.00 | 4/30/2011         | Climaticas | 718,101.68  | 9,665,657.10 |
| 2011 | MAYO             | 521.00 | 5/3/2011          | Red        | 721,732.52  | 9,658,279.89 |
| 2011 | MAYO             | 521.00 | 5/9/2011          | Climaticas | 727,416.00  | 9,646,641.85 |
| 2011 | MAYO             | 521.00 | 5/19/2011         | Red        | 718,286.94  | 9,661,729.68 |
| 2011 | MAYO             | 521.00 | 5/19/2011         | Red        | 717,438.68  | 9,673,685.19 |
| 2013 | "                | "      | "                 | "          | "           | "            |
| 2014 | "                | "      | "                 | "          | "           | "            |
| 2015 | DICIEMBRE        | 521.00 | 12/31/2015        | Climaticas | 730,941.98  | 9,647,892.93 |

**ANEXO C****CODIGO DEL MODELO COMPUTACIONAL DE PREDICCIÓN DE FALLAS****DESCRIPCIÓN DE LAS LIBRERÍAS**

#Se importan las librerías necesarias  
 #Las no instaladas se pueden instalar mediante el comando pip install- <<librería>> en el CMD  
 #Numpy es una librería de contenedores multidimensionales de datos genéricos  
 #Matplotlib es la librería encargada de los plots o graficas  
 #Pandas es el lector de datos y archivos  
 #Keras es el modulo exportado que corre sobre la librería de deep learning TENSORFLOW  
 #Sklearn es la librería encargada de los procesos de machine learning

**IMPORTACIÓN DE LAS DEPENDENCIAS NECESARIAS**

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz



```
import numpy
import matplotlib.pyplot as plt
import pandas
import math
```

```
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout, Activation
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
```

```
.....
.....
#Del conjunto de datos se determina, las dimensiones a considerar para el problema
#dependiendo las columnas que van a ser analizadas
```

```
def create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset) - look_back - 1):
        a = dataset[i:(i + look_back), :]
        dataX.append(a)
```

```
        #Aquí se especifica el número de características de entrada, en este caso 20.
        #Python entiende los paréntesis como "numero de" y los corchetes como "posición
en"
        #En este caso 19 es la posición 20, porque el contador inicia en 0
```

```
        dataY.append(dataset[i + look_back, 19])
    return numpy.array(dataX), numpy.array(dataY)
```

```
.....
.....
#Se otorga el valor del random seed (número semilla aleatorio), define la pronosticabilidad

numpy.random.seed(7)
```

## LECTURA DE LOS DATOS DE ENTRADA

```
.....
# Con la librería pandas se leen los datos en el .csv cargados en el notebook
```

```
dataframe = pandas.read_csv('20.csv', engine='python')
dataset = dataframe.values
```

## NORMALIZACIÓN

```
.....
#Normalizar el conjunto de datos en valores entre 0 y 1
```

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz



```
scaler = MinMaxScaler(feature_range=(0, 1))  
dataset = scaler.fit_transform(dataset)
```

```
.....  
.....  
# Establecemos el porcentaje de datos a entrenar
```

### AJUSTE DE ENTRENAMIENTO Y PRUEBA

```
.....  
# separamos los datos en porcentajes de entrenamiento 70 por ciento  
# restamos de la totalidad 100%-70% = 30% para testeo o pruebas
```

```
train_size = int(len(dataset) * 0.70)  
test_size = len(dataset) - train_size  
train, test = dataset[0:train_size, :], dataset[train_size:len(dataset), :]
```

```
# da la nueva forma X=t and Y=t+1, hace referencia al valor después del tiempo t
```

```
look_back = 20  
trainX, trainY = create_dataset(train, look_back)  
testX, testY = create_dataset(test, look_back)
```

```
# La nueva forma seria [muestras, saltos de tiempo, características]
```

```
trainX = numpy.reshape(trainX, (trainX.shape[0], look_back, 20))  
testX = numpy.reshape(testX, (testX.shape[0], look_back, 20))
```



.....  
**DESCRIPCIÓN DE LA RED**

.....  
# Long-short term memory, o LSTM es RMLP (neuronas de memoria de larga plazo)  
#El validation split determinada el porcentaje de datos que van a ser usados para validación  
#Epochs es una palabra que puede traducirse como época, relacionado con el contexto temporal  
#se puede entender como las Iteraciones (un barrido a todos los datos)  
# El número batch (datos para aprendizaje por lotes) está dentro de cada uno de los epochs  
#Mientras más grande el epoch y el batch, más largo el entrenamiento pero no necesariamente mejor

.....  
**DISEÑO DE LA RED NEURONAL**

.....  
#Crea la red neuronal -long short term memory-  
#Se apila capas con model.add.

```
model = Sequential()  
model.add(LSTM(150, input_shape=(look_back,20),return_sequences=True))  
model.add(LSTM(75, input_shape=(look_back,20),return_sequences=False))  
  
#si se desea una red más profunda por ejemplo, entonces:  
#model.add(Dense(16, kernel_initializer='uniform', activation='relu'))  
#model.add(Dense(8, kernel_initializer='uniform', activation='relu'))  
#model.add(Dense(4, kernel_initializer='uniform', activation='relu'))  
#model.add(Dense(2, kernel_initializer='uniform', activation='relu'))  
  
model.add(Dense(1, kernel_initializer='uniform', activation='linear'))  
model.compile(loss='mean_squared_error', optimizer='adam')  
history= model.fit(trainX, trainY, validation_split=0.1, epochs=150, batch_size=2,  
verbose=0.25)
```

.....  
**GRÁFICA DE LAS PERDIDAS VS LAS ITERACIONES**

.....  
#Grafica la perdida 'error vs epocas de entrenamiento'  
`plt.plot(history.history['loss'],color='blue')`  
`plt.plot(history.history['val_loss'],color='black')`  
`plt.title('model loss')`  
`plt.ylabel('pérdida')`  
`plt.xlabel('época')`  
`plt.legend(['entrenamiento', 'validación'], loc='upper right')`  
`plt.show()`

.....  
**REALIZACIÓN DE LAS PREDICCIONES**

.....  
# realizar las predicciones  
`trainPredict = model.predict(trainX)`  
`testPredict = model.predict(testX)`

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz



.....  
**DESNORMALIZACIÓN DE LAS PREDICCIONES**  
.....

#Muestra las predicciones en número real. Des-normaliza lo que antes se normalizó  
#Crea una matriz vacía con las dimensiones de la entrada

```
trainPredict_extended = numpy.zeros((len(trainPredict),20))
```

#En la matriz creada coloca las predicciones

```
trainPredict_extended[:, 19] = trainPredict[:,0]
```

#Desnormaliza y pone ahí la columna que queremos predecir

```
trainPredict = scaler.inverse_transform(trainPredict_extended)[:, 19]  
print(trainPredict)
```

#Crea una matriz vacía con las dimensiones de la entrada

```
testPredict_extended = numpy.zeros((len(testPredict),20))
```

#En la matriz creada coloca las predicciones

```
testPredict_extended[:, 19] = testPredict[:,0]
```

#Des-normaliza y pone ahí la columna que queremos predecir para luego mostrala

```
testPredict = scaler.inverse_transform(testPredict_extended)[:, 19]
```

```
print(testPredict)
```

.....  
.....  

```
trainY_extended = numpy.zeros((len(trainY),20))
```

```
trainY_extended[:, 19]=trainY
```

```
trainY=scaler.inverse_transform(trainY_extended)[:, 19]
```

```
testY_extended = numpy.zeros((len(testY),20))
```

```
testY_extended[:, 19]=testY
```

```
testY=scaler.inverse_transform(testY_extended)[:, 19]
```

.....  
**CÁLCULO DEL ERROR**  
.....

# calcula el error medio cuadratico de la raíz

```
trainScore = math.sqrt(mean_squared_error(trainY, trainPredict))
```

```
print('Train Score: %.2f RMSE' % (trainScore))
```

```
testScore = math.sqrt(mean_squared_error(testY, testPredict))
```

```
print('Test Score: %.2f RMSE' % (testScore))
```

#Transforma el entrenamiento de predicciones para el ploteo

Juan Fernando Segarra López

Pedro Sebastián Andrade Veloz



```
trainPredictPlot = numpy.empty_like(dataset)
trainPredictPlot[:, :] = numpy.nan
trainPredictPlot[look_back:len(trainPredict)+look_back, 19] = trainPredict
```

# Transforma el entrenamiento de predicciones para el ploteo

```
testPredictPlot = numpy.empty_like(dataset)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, 19] = testPredict
```

## GRÁFICA DE LOS RESULTADOS

# Grafica los datos reales, predicciones y entrenamiento vs el tiempo//365 días (un año)//

```
serie,=plt.plot(scaler.inverse_transform(dataset)[:, 19],color='black',label='datos')
prediccion_entrenamiento,=plt.plot(trainPredictPlot[:, 19],color='blue',label='entrenamiento',linestyle='--')
prediccion_test,=plt.plot(testPredictPlot[:, 19],color='red',label='predicciones',linestyle='--')
plt.title('Fallas')
plt.ylabel('Coordenada X')
plt.xlabel('Día del Año')
plt.legend(loc='upper left')
plt.show()
```