

Urkund Analysis Result

Analysed Document: Tesina_final_Versio_Urkund (3).docx (D42635005)
Submitted: 10/16/2018 6:44:00 PM
Submitted By: xiomaraonofreespin@hotmail.com
Significance: 3 %

Sources included in the report:

urkund onofre_campoverde.docx (D42496385)

Instances where selected sources appear:

5

INTRODUCCIÓN

El presente trabajo de investigación se lleva a cabo con el propósito de realizar un análisis predictivo de datos utilizando redes neuronales artificiales, con el

uso de las mismas se podrá predecir resultados con la utilización del software Matlab. El caso de estudio se desarrollará con datos obtenidos de la empresa CNEC de la ciudad de Milagro basándose en los consumidores de energía por mes y con el grupo de consumo (residencial, industrial, comercial).

Las redes neuronales artificiales utilizan su capacidad de aprendizaje adaptativo para auto organizar la información que reciben durante el aprendizaje y / o la operación, que consiste en modificar la red neuronal completa para llevar a cabo un objetivo específico, en el caso de la presente investigación.

El análisis predictivo de los datos de los diferentes grupos de consumo. Al usar Matlab, se llevará a cabo un análisis de datos predictivo, ya que este software permite entrenar redes neuronales con el algoritmo de propagación hacia atrás feedforward, para resolver problemas específicos. Además, permite entrenar su propia red neuronal artificial desde cero o usar un modelo previamente entrenado para realizar la transferencia de aprendizaje.

Para la predicción de nuestro trabajo de investigación se propone una red neuronal donde la entrada es el número de consumidores y la salida son los tres grupos de consumo. En primera instancia se genera un archivo con información de la demanda en kilovatios, cabe destacar que los datos deben estar debidamente normalizados, para que la salida de la información se adapte a los valores comprendidos entre 0 y 1.

Es así que en el Capítulo 1 se presenta la problemática a estudiar, la cual se centra en el análisis predictivo de datos mediante

el uso del software Matlab.

En el Capítulo 2 abarca la recopilación de la información de diferentes repositorios académicos como revistas científicas y libros que son necesario para el presente para el desarrollo del marco teórico. En

el Capítulo 3, se procede a presentar la metodología utilizada para el desarrollo del presente proyecto

experimental.

En

el Capítulo 4-5, se presenta paso a paso la ejecución y elaboración de las redes neuronales artificiales en el software Matlab.

Al final se presentaran los resultados de la investigación con su respectivo análisis, presentando

en el capítulo 5

conclusiones obtenidas durante el desarrollo de las redes neuronales artificiales para el análisis predictivo de datos.

CAPITULO 1 EL PROBLEMA 1.1 Planteamiento del problema

El campo de la tecnología crece de manera acelerada y con mucha complejidad, en la que se involucra el análisis predictivo de datos, un término utilizado para referirse al conjunto de procesos que involucran la aplicación de diferentes técnicas computacionales para hacer predicciones sobre el futuro; Basado en Comment11 datos pasados.

Existen diferentes modelos Comment12 predictivos de datos

como las redes neuronales, aprendizaje automático, big data y conmutación cognitiva;

que aplican los resultados conocidos que se ejecutan para predecir valores, con datos diferentes o completamente nuevos, en un proceso repetitivo, esto proporciona resultados en forma de predicciones representadas por el grado de probabilidad de la variable objetivo.

Uno de estos modelos de datos predictivos son las redes neuronales, que imitan el comportamiento del cerebro para desarrollar tareas específicas, basadas en el reconocimiento de patrones y permiten modelar gráficamente los parámetros del estudio. Estas redes se conectan entre sí y trabajan juntas, aprendiendo sobre el proceso, que cuando se combina ayuda a predecir ciertos resultados, dando lugar al problema de encontrar la combinación que mejor se adapte a la formación de la red neuronal, marcando así la precisión de la red neuronal del algoritmo ejecutado.

En los últimos años, la computación paralela y las redes neuronales han despertado gran interés en los campos de la computación y la inteligencia artificial. Anteriormente, los métodos y herramientas informáticos utilizados para procesar información se caracterizaban, en general, por representar explícitamente el conocimiento a través de reglas, redes semánticas y modelos probabilísticos, entre otros; centrar la atención en las causas que intervinieron en el problema y en sus relaciones; y para procesar la información secuencialmente.

Dentro de las aplicaciones de las redes neuronales artificiales, hay predicciones, en áreas de finanzas, economía, agricultura, políticas, condiciones meteorológicas, etc. La elaboración de una red neuronal implica definir el número de capas ocultas y el número de neuronas en ellas, el tipo de conexiones entre neuronas, la función de transferencia o respuesta de las neuronas y el mecanismo de aprendizaje,

es decir, las redes neuronales mediante su aprendizaje, que lo realiza de manera interno, permite predecir un resultado dándole solución a una problemática de estudio.

En

el presente trabajo se aplicará la técnica de propagación hacia atrás que consiste en utilizar una optimización Comment13 basada en derivadas, donde el error es derivable no solo en términos de los pesos de la capa de salida, pero también en función de los pesos de la capa oculta.

1.2

Línea de Investigación

Desarrollo e innovación tecnológica

1.3 Objetivos 1.3.1 Objetivo General Realizar análisis predictivo de datos obtenidos en la empresa CNEL de la ciudad de Milagro en el consumo de energía residencial, comercial e industrial, mediante el uso de redes neuronales artificiales con la utilización del software MATLAB Comment21 Comment22 .

1.3.2 Objetivos Específicos

- Ajustar los valores de pesos, umbrales, número de neuronas y coeficiente de aprendizaje para alcanzar coeficiente R cercano a 1.
- Utilizar la interfaz nntool para ingreso de datos y entrenamiento de la red.
- Normalizar los datos de consumo de energía.
- Analizar los resultados obtenidos y realizar predicciones en base a una estimación de incremento o disminución de clientes.

1.4

Justificación

El análisis predictivo de datos se puede usar ampliamente en casi cualquier sector, no solo en negocios, detectar oportunidades comerciales, detectar y reducir el fraude, la retención de clientes, predecir fallas en los sistemas, sino también en otros campos en los que todos nos beneficiamos de cómo detectar el cáncer en los pacientes. Evolución de las epidemias, ahorro de costos en las organizaciones públicas, reconocimiento de voz, la lista es interminable.

Las redes neuronales artificiales (RNA) han demostrado ser técnicas que se pueden usar para determinar las relaciones entre las variables de entrada y salida, que luego permiten predecir resultados en función de los valores de las variables de entrada. Estas características de los RNA se han aplicado en diferentes áreas del conocimiento, especialmente para comparar patrones y clasificación.

Cada vez más, las RNA se utilizan como herramientas para realizar regresiones no lineales y para desarrollar modelos con relaciones variables muy complejas. Una de las ventajas del RNA con respecto a los métodos estadísticos convencionales es que no requieren suposiciones acerca de la distribución de los datos a analizar. Existen diferentes problemas en los procesos

alimentarios y los bioprocesos que no se pueden caracterizar o resolver utilizando aproximaciones con modelos físicos. En estas situaciones, el modelado con RNA se puede utilizar como alternativas potenciales.

•

CAPITULO 2 MARCO TEÓRICO 2.1 Neuronas biológicas

Las señales encontradas en una neurona biológica son de naturaleza eléctrica y química. La señal generada por la neurona y transportada a lo largo del axón es eléctrica, mientras que la señal que se transmite entre los terminales del axón de una neurona y las dendritas de las siguientes neuronas es de origen químico. Específicamente, se realiza mediante neurotransmisores que fluyen a través de una región especial, llamada sinapsis que se encuentran entre los terminales de los axones y las dendritas de las siguientes neuronas, CITATION Edu17 \p 20 \t \l 3082 (Caicedo & Lopez, 2017, pág. 20).

Figura 11. La neurona biológica

Fuente: Caicedo & López, 2017

Para establecer una similitud directa entre la actividad de una neurona biológica y una neurona artificial, analicemos los siguientes aspectos funcionales: 1. Los elementos del proceso (neurona), reciben señales de entrada

debido a la gran capacidad de procesamiento y complejidad que tienen. 2.

Las señales pueden ser modificadas por los pesos sinápticos: la comunicación

que se genera entre las neuronas se realiza por medio de la sinapsis. 3. Los elementos de los procesos suman las entradas afectadas por la sinapsis, para definir qué hace la neurona en base a las señales de acumulación que se suministran en ella. 4. Las neurona generan una señal de salida, si la señal recibida es grande esta genera la activación de la neurona caso contrario esta seguirá inactiva.

5.

La salida del elemento del proceso puede ir a muchas neuronas, cuando una neurona se activa, es capaz de transmitir un impulso eléctrico a las neuronas con las que tiene contacto. CITATION Edu17 \p 20-21 \t \l 3082 (Caicedo & Lopez, 2017, págs. 20-21).

2.2 Redes neuronales artificiales 2.2.1 Concepto

Existen varios conceptos sobre redes neuronales artificiales (ARN), considerando que para Montaña, CITATION Mon12 \n \t \l 3082 (2012) son sistemas de procesamiento de información cuya estructura y funcionamiento están inspirados en redes neuronales biológicas. Consisten en un conjunto de elementos de procesamiento simples llamados nodos o neuronas conectadas entre sí por conexiones que tienen un valor numérico modificable denominado peso.

Según

CITATION Pil17 \l 3082 (Piloto, 2017) CITATION ESC14 \l 3082 (ESCOM, 2014),

los ARN consisten en unidades de procesamiento que intercambian datos o información. Se utilizan para reconocer patrones, incluyendo imágenes, manuscritos, secuencias de tiempo, tendencias financieras, etc. Una de sus características fundamentales es que tienen la capacidad de aprender y mejorar su funcionamiento.

Figura 22. Neuronas Artificiales

Fuente: Piloto, 2017

2.2.2

Tipos de redes neuronales 2.2.2.1 Según la Topología de redes neuronales

La topología es la cantidad de elementos de procesamiento que componen la red y las interconexiones entre ellos. Los elementos de procesamiento de una red neuronal se distribuyen por capas. La cantidad de elementos de procesamiento puede variar en algunas capas con respecto a otras. En general, cuanto mayor sea la diferencia entre los elementos de entrada y los elementos de salida, será necesario proporcionar a la estructura más capas ocultas. Estas capas ocultas crean una representación interna de los patrones de entrada. La capacidad de la red para procesar información crece en proporción directa a la cantidad de capas ocultas,

CITATION ESC14 \l 3082 (ESCOM, 2014).

Para LópezCITATION Gus18 \n \t \l 3082 (2018),

una topología determina la cantidad de nodos que se conectarán, el método de acceso múltiple, el tiempo de respuesta, la velocidad de la información, el costo, el tipo de aplicaciones, etc.

Los tipos de redes neuronales en cuanto a su topología según Martínez & Serrano CITATION Mar12 \n \t \l 3082 (2012), pueden ser:

- Redes de monocapa: Tiene dos niveles de neuronas, la primera asociada con el vector de entrada sin ningún procesamiento de estos datos, por esta razón no se considera formalmente como una capa, solo sirve como una interfaz entre los datos de entrada y las siguientes capas de neuronas. El primer nivel tiene la misma dimensión del vector de entrada, ingresa en el mismo nivel y los datos se transfieren al siguiente nivel, modificado por los pesos sinápticos donde solo una capa de procesamiento está disponible dentro de su arquitectura, de ahí su nombre.

Figura 33. Red Monocapa

Fuente: Martínez & Serrano, 2012

•

Redes multicapa: En este caso, la red tiene un nivel de entrada con n neuronas y una capa de salida de m neuronas; cuyo comportamiento es similar al descrito en la red de monocapa. La diferencia sustancial es que se incluye una nueva capa intermedia entre la entrada y la salida, esta capa se llama capa oculta, que está formada por neuronas h .

Figura 44.Red Multicapa

Fuente: Martínez & Serrano, 2012

•
Redes Feedforward: En este tipo de red neuronal artificial, la información fluye en una sola dirección desde las neuronas de entrada a la capa o capas de procesamiento, para los casos de redes monocapa y multicapa, respectivamente; Hasta llegar a la capa de salida de la red neuronal.

Figura 55.Red Feedforward Fuente: Martínez & Serrano, 2012

• Redes recurrentes: En este tipo de redes neuronales, la información no siempre fluye en una dirección, ya que puede realimentarse en capas anteriores a través de conexiones sinápticas. Este tipo de red neuronal puede ser monocapa o multicapa. Un grupo de neuronas, después de recibir la información que proviene de las unidades de retardo, llevan su señal de estímulo a todas las neuronas de salida. Este conjunto de neuronas se suele llamar neuronas de contexto.

Figura 66.Redes Neuronales Recurrentes

Fuente: Martínez & Serrano, 2012

2.2.2.2

Perceptrón

Para Moreno CITATION Dar16 \n \t \l 3082 (2016), consiste en una suma de las señales de entrada, multiplicadas por pesos seleccionados al azar. La entrada se compara con un patrón preestablecido para determinar la salida de la red. Si en la comparación, la suma de las entradas multiplicadas por los pesos es mayor o igual que el patrón preestablecido, la salida de la red es una, de lo contrario la salida es cero (0).

Figura 77.Red Perceptrón Fuente: Moreno, 2016

2.2.2.3 Adaline

De acuerdo a Tanco CITATION Fer15 \n \t \l 3082 (2015), es similar al Perceptrón, excepto en su función de transferencia, que es una función lineal en lugar de un fuerte limitador como en el caso del Perceptrón, presenta la misma limitación, en términos del tipo de problemas que se pueden resolver, ambas redes solo puede resolver problemas linealmente separables. Sin embargo, el algoritmo LMS es más poderoso que la regla de aprendizaje Perceptrón, ya que

minimiza el error cuadrático medio, una característica que lo hace bastante práctico en aplicaciones de procesamiento de señales digitales, por ejemplo, las líneas telefónicas de larga distancia usan la red Adaline para cancelar el ruido inherente a su ruta.

Figura 88.Red Adaline

Fuente: Tanco (2015)

2.2.2.4

Backpropagation

Según Matich CITATION Dam15 \n \t \l 3082 (2015), es un tipo de red de estudio inspeccionado, que genera un ciclo

de difusión-ajuste de 2 fases. Cuando el

patrón

se ha adaptado a la entrada de la red, este se difunde desde la primera capa a través

de la capa superiores, hasta lograr una salida, esta salida esperada se comprueba con la anhelada y se realiza cálculos de error por las salidas que se generen.

Figura 99.Backpropagation Fuente: Matich, 2015

Además, Boza CITATION Edw14 \n \t \l 3082 (2014) presenta que la importancia de este método, es por medio del entrenamiento de la red, las neuronas

adquieren conocimiento y características mediante la fase de preparación, luego de esto las neuronas generan resultado en base a los conocimientos adquiridos por medio del entrenamiento.

2.2.3

Función de Activación de las Neuronas 2.2.3.1 Hardlim La función de transferencia lleva la salida de la red a cero, se da si el argumento de la función es menor que cero y la toma a uno si este argumento es mayor que uno. Esta función crea neuronas que clasifican las entradas en dos categorías diferentes, una característica que les permite ser utilizadas en la red de tipo perceptrón, CITATION Car172 \l 3082 (Espino & Martínez, 2017).

Ecuación:

Figura 1010.Función Hardlim

Fuente: Espino & Martínez, 2017

2.2.3.2

Modificación de Hardlim La modificación de esta función se puede ver en la restricción del espacio de salida a valores entre 1 y -1, como se muestra en el gráfico, CITATION Car172 \l 3082 (Espino & Martínez, 2017) Ecuación:

Figura 1111.Modificación de Hardlim

Fuente: Espino & Martinez,2017

2.2.3.3

Purelin La salida de una función de transferencia lineal es igual a su entrada, puede ver las características de la salida a la red, en comparación con la entrada, más un valor de ganancia, las neuronas que utilizan esta función de transferencia se utilizan en el tipo de red Adaline, CITATION Car172 \l 3082 (Espino & Martínez, 2017). Ecuación: $y = a$

Figura 1212.Función Purelin

Fuente: Espino & Martinez,2017

2.2.3.4

Sigmoidal (logsig) Esta función toma los valores de entrada, que pueden variar entre más y menos infinito, y restringe la salida a valores entre cero y uno. Se usa comúnmente en redes multicapa, como Backpropagation, en parte porque la función logsig es diferenciable, CITATION Car172 \l 3082 (Espino & Martínez, 2017).

Ecuación:

Figura 1313.Función Sigmoidal (logsig)

Fuente: Caballero, Díaz, Moradillos, & Pavoni, 2014

La función logarítmica sigmoidal es ampliamente utilizada como una función de la activación de las neuronas que componen una red neuronal artificial. Debido al uso de la función sigmoidal como una función de activación en redes neuronales y el paralelismo de estas redes, una de las formas más utilizadas es su implementación en hardware reconfigurable.

CITATION Cab14 \l 3082 (Cabellero, Diaz, Moradillos, & Pavoni, 2014).

2.2.4 Topología de redes neuronales 2.2.4.1 Número de Capas

Los elementos que poseen las redes neuronales son asociados a través de capas dentro de un conjunto de neuronas, en base a la localización de las capas de la red neuronal, adquiere distintos

nombres:

- Capa de entrada: Recibe señales de la entrada de la red; algunos autores no consideran el vector de entrada como una capa porque no se lleva a cabo ningún proceso allí.
- Capas

ocultas: Estas capas son aquellas que no tienen contacto con el entorno externo, sus elementos pueden tener diferentes conexiones y esto es lo que determina las diferentes topologías de la red. • Capa de salida: Recibe información de la capa oculta y transmite la respuesta a la externa ambiente, CITATION Edu17 \l 3082 (Caicedo & Lopez, 2017).

2.2.4.2 Número de neuronas por capas

Busca analizar qué se refiere a cuántos niveles o capas, así como el número de neuronas por capas, son necesarios. En el nivel inicial, hay una unidad de sensor para cada función de entrada a la red, ya que muchas neuronas se colocan en la capa de salida según sea necesario. Se ha comprobado que dos capas ocultas son suficientes para resolver cualquier problema. No hay criterios rigurosos para determinar el número de neuronas en las capas ocultas, CITATION Edu17 \l 3082 (Caicedo & Lopez, 2017).

2.2.4.3 Grado de Conectividad

La conectividad

que se genera entre los nodos de una red está vinculada con

las salidas de las neuronas estas se encaminan para transformarse en entradas de otras neuronas. La señal de salida de un nodo consigue ser la entrada de otra neurona o incluso de sí misma (conexión de corriente automática). Cuando no hay salida de las neuronas se ingresa desde las neuronas del mismo nivel o niveles anteriores pero están conectadas a las neuronas del siguiente nivel, hablamos de red de propagación directa, o red de alimentación previa o de alimentación progresiva. La estructura es la de un grafo dirigido acíclico (GAD), CITATION Edu17 \l 3082 (Caicedo & Lopez, 2017).

2.2.4.4 Tipo de conexiones entre neuronas

Las variantes de aquellos

que contienen una sola capa de lógica son bucles y retroalimentación de capas complejas. Estos algoritmos hace uso de la lógica de programación para saber la vigilancia de las funciones de las organizaciones en base a sus funciones, por lo general la mayor parte de los sistemas hacen uso de pesos para la transformación de los medidas de rendimiento y las distintas conexión que se dan entre las neuronas. Las redes neuronales se consideran redes autónomas ya que adquieren aprendizaje por medio de

las reglas de entrada descritas. CITATION Edu17 \l 3082 (Caicedo & Lopez, 2017).

2.2.5 Entrenamiento de redes neuronales

En cuanto al entrenamiento de redes neuronales artificiales, para Coronel CITATION Bry16 \n \t \l 3082 (2016) el entrenamiento de una red neuronal multicapa se realiza a través de un proceso de aprendizaje, para realizar este proceso primero debe haber definido la topología de la red que es: número de neuronas en la capa de entrada que depende de la cantidad de componentes de la entrada de vector, número de capas ocultas y número de neuronas de

cada una de ellas, número de neuronas en la capa de salida que depende del número de componentes de vectores de salida o patrones de destino y funciones de transferencia requeridas en cada capa, según la topología elegida, valores iniciales se asignan a cada uno de los parámetros que componen la red, CITATION Edu17 \l 3082 (Caicedo & Lopez, 2017).

2.2.5.1

Aprendizaje por esfuerzo

En el aprendizaje por refuerzo, se indica al algoritmo si la decisión adoptada es buena o no. Se tiene, por tanto, una respuesta global que se provee. Por el contrario, el algoritmo no sabe exactamente lo que tendría que haber decidido. En el caso de las redes neuronales, se utiliza a menudo esta forma de aprendizaje cuando se busca obtener comportamientos complejos que hacen intervenir varias decisiones. Es el caso, por ejemplo, en robótica, para crear adversarios inteligentes en los videojuegos. En efecto, se busca un programa capaz de tomar las distintas decisiones para alcanzar una posición ganadora, CITATION Cev18 \l 3082 (Cevallos, 2018).

2.2.5.2 Aprendizaje supervisado

Este aprendizaje su característica principal está centrado en el enfoque de entrenamiento este es supervisado por un agente externo, en el mundo real los problemas son las medidas de entrada y salida con la que se alimenta la red neuronal, quien opere el sistema es el guía el cual le brinda los conocimiento para el enseñanza que la red

CITATION Cev18 \l 3082 (Cevallos, 2018).

Los datos de entrada se le presentan al supervisor y a la red de neuronal simultáneamente, el supervisor propone la salida deseada para ser comparada con la salida de la red neuronal artificial. Para que la labor del supervisor sea exitosa se define en él, un error de entrenamiento como la diferencia entre la salida deseada o esperada y la salida que produce la RNA CITATION Cev18 \l 3082 (Cevallos, 2018).

Función:

Figura 1414. Aprendizaje supervisado Fuente: Cevallos, 2018

Donde: y : Salida de la red neuronal artificial d : Salida deseada

2.2.5.3 Descenso por gradiente El método de descenso por gradiente se aplica únicamente a redes monocapa de tipo perceptrón (con una función heavyside). Los pesos se optimizan en varios pasos en el conjunto del aprendizaje Consideremos una red que posea entradas, x y ejemplos. Se denomina \hat{y} la salida obtenida sobre el n ésimo ejemplo, e a la salida esperada CITATION Cev18 \l 3082 (Cevallos, 2018). El error cometido sobre un punto se expresa, entonces: $= \hat{y} - e$

Al principio de cada paso, se inicializa a 0 las modificaciones a aplicar a los pesos w .

La variación se denomina Δ . Con cada ejemplo probado, se modificará de la siguiente manera:

$$\Delta w = \Delta w + (\text{error}_{\text{salida}} - \text{error}_{\text{oculta}}) \cdot \text{activación}$$

2.2.5.4 Retropropagación

Los métodos anteriores se aplican únicamente a los perceptrones. Sin embargo, en muchos casos, se utilizará redes con capas (con una función de activación sigmoide). Existe un aprendizaje posible: por propagación hacia atrás (llamado "Backpropagation" en inglés). Se van a corregir, en primer lugar, los pesos entre las neuronas de salida y las neuronas ocultas, y a continuación se propagará el error hacia atrás (de ahí el nombre) y corregir los pesos entre las neuronas ocultas y las entradas. Esta corrección se hará tras cada ejemplo, y tendrá varios pasos (con órdenes diferentes si es posible) para converger hacia un valor óptimo CITATION Cev18 \l 3082 (Cevallos, 2018).

Las fórmulas que usa el algoritmo backpropagation o también conocido como programación hacia atrás se encuentran:

- Descenso del gradiente, su fórmula es: $w = w - \eta \cdot \text{error} \cdot \text{activación}$
- Función de activación $f(x) = \frac{1}{1 + e^{-x}}$ Donde la verifica que la derivada sea igual a la función: $f'(x) = f(x) \cdot (1 - f(x))$

- La regla de la cadena nos permite realizar las derivadas de las funciones.

$$\frac{\partial \text{Error}}{\partial w} = \sum_i \frac{\partial \text{Error}}{\partial \text{activación}_i} \cdot \sigma'_i \cdot w$$

2.6 Análisis predictivo de datos

El análisis de predicción de la información está vinculada a la minería de datos debido que esta abarca la extracción de los datos para generar predicciones y tendencia en base al comportamiento, estas predicciones pueden ser dada para un determinado tiempo

0: urkund onofre_campoverde.docx

84%

ya sea pasado, presente o futuro. El análisis predictivo se centra en la identificación

de variables que se hayan suscitado en el pasado para de esta manera realizar una predicción a futuro. La precisión de los resultados que se generan para a tener una dependencia del análisis que se realice a la información, del mismo modo será la calidad en base a

los supuestos CITATION Car172 \l 3082 (Espino & Martínez, 2017).

El análisis predictivo usa datos históricos para predecir eventos futuros. Normalmente, los datos históricos se utilizan para crear un modelo matemático que captura tendencias importantes. Este modelo predictivo se utiliza luego con los datos actuales para predecir lo que sucederá a continuación, o sugerir acciones para llevar a cabo con el fin de obtener resultados óptimos. El análisis predictivo ha recibido mucha atención en los últimos años debido a los avances en la tecnología que lo soporta, especialmente en las áreas de big data y machine learning, CITATION Car172 \l 3082 (Espino & Martínez, 2017).

2.6.2

Aprendizaje computacional

El aprendizaje computacional es vital en el análisis de predicciones, este por medio de estudio y técnicas de datos logra el descubrimiento de las relaciones que se dan entre las variables, esta desde el comienzo puede que no sean de importancia es decir se las considere pocos importantes, pero en abe a los análisis estas variables puede ser técnicas de trascendencia

CITATION Car172 \l 3082 (Espino & Martínez, 2017).

2.7 Programa matemático de aprendizaje 2.7.1 Matlab

Actualmente, las empresas están encontrando formas innovadoras de aplicar análisis predictivo

a través de MATLAB® para crear nuevos productos y servicios, así como para resolver problemas de por vida de nuevas maneras, que es un entorno de cálculo técnico de alto rendimiento para el cálculo numérico y la visualización CITATION Jos141 \l 3082 (Sánchez, 2014). Integra: • Análisis numérico • Cálculo de matriz • Procesamiento de señales • Gráficos

Es fácil de usar, donde los problemas y soluciones se expresan a medida que se escriben matemáticamente, sin programación tradicional. MatLab fue escrito originalmente para proporcionar un acceso fácil al software matricial desarrollado por los proyectos LINPACK y EISPACK, que juntos representan los programas de cálculo matricial más avanzados, CITATION Jos141 \l 3082 (Sánchez, 2014).

Es un sistema interactivo cuyo elemento básico de datos es una matriz que no requiere dimensionamiento. Esto permite resolver muchos problemas numéricos en una fracción del tiempo que llevaría hacerlo en lenguajes como C, BASIC o FORTRAN. MATLAB ha evolucionado en los últimos años gracias a la colaboración de muchos usuarios, CITATION Jos141 \l 3082 (Sánchez, 2014).

La principal ventaja de MATLAB es el uso de familias de comandos de áreas específicas llamadas cajas de herramientas. Lo más importante para los usuarios de MATLAB es que las cajas de herramientas les permiten aprender y aplicar la teoría. Las cajas de herramientas son grupos de comandos de MATLAB (archivos M) que amplían el entorno de MATLAB para resolver problemas en áreas específicas de la ciencia y la ingeniería. Por ejemplo, hay cajas de herramientas para las áreas de procesamiento digital de señales, sistemas de control, redes neuronales, lógica difusa, ondículas, etc. CITATION Jos141 \l 3082 (Sánchez, 2014).

CAPÍTULO 3 METODOLOGÍA

1. 2. 3. 3.1 Metodología Experimental Según CITATION Ker79 \l 3082 (Kerlinger, 1979) indica que el estudio experimental se manifiesta cuando el investigador trata la variable independiente de manera intencional, es decir que los resultados finales son provenientes de una causa. Un ejemplo claro seria nuestra investigación que se va analizar datos provenientes de una empresa y mediante software, se encontrará resultados factibles para una posible

solución a la problemática. En cambio, CITATION Sam98 \l 3082 (Sampieri, 1998), relata que un estudio para que tenga un fundamento y sea aceptado tiene que poseer una sustentación de la misma, en ocasiones nos pronunciamos ante un tema pero no tenemos fundamentación y es ahí donde nuestra palabra queda en incertidumbre por parte de nuestros receptores. Son muchos los autores que mencionan la definición de investigación experimental, todos dan diferentes puntos de vista pero a sí mismo todos se dirigen a que la investigación experimental depende del tipo de investigación y sobre el alcance del mismo. En nuestro tema se recolectará información y dicha información será analizada, tabulada y por último sus conclusiones con sus respectivas justificaciones. Cabe recalcar que los datos que se van a hacer analizados son datos verídicos de una empresa. Ellos, CITATION Sam00 \l 3082 (Sampieri, Fernandez, & Baptista, 2000), comentan que tanto la investigación experimental como la no experimental son de gran aporte al conocimiento y al momento de elegir una u otra, sus resultados serán apropiados, de gran importancia y validez para las personas que estén interesadas en estudios con investigación experimental. Este trabajo posee una metodología experimental, por lo tanto se

realizará una predicción a futuro sobre la problemática de consumo de energía, teniendo en cuenta que la empresa como referencia es CNEL.

3.2

Preparación de los datos Una vez que la empresa CNEL (Corporación Nacional de Electricidad) de la Ciudad de Milagro nos facilitó la información necesaria para el desarrollo de nuestra investigación experimental, en base al consumo de energía del año 2017-2018, en 3 diferentes tipos de consumo: residencial, industrial y comercial. Luego con los datos obtenidos se establecen las entradas de la red neuronal, dichas entradas son los tipos de consumo de energía, es decir, nuestra red neuronal tendrá

una entrada que es el número de clientes y tres salidas en sectores de consumo: residencial, industrial y comercial, por lo tanto

0: urkund onofre_campoverde.docx

86%

el mes de consumo se lo conoce como conjunto de entrenamiento. Se ingresarán datos de dos años, a cada conjunto de valores ingresados

en la red neuronal multicapa

son conocidos como épocas, el perceptrón adquiera aprendizaje por medio de la información proporcionada. Una vez que la neurona posea conocimiento se procederá a realizar un test de validación, que consiste en el ingresar más información sobre el consumo de energía para luego ser comparado con los datos ya suministrado en la neurona.

Cuando se obtiene los resultados, se los clasifica

0: urkund onofre_campoverde.docx

92%

y se procede a utilizar la siguiente fórmula que consiste en la división del número de consumo por mes, entre el total

de salidas que generó la neurona y el resultado de la división será multiplicado por 100
Comment93 ,

esta fórmula nos dará la probabilidad a predecir del consumo de energía por mes.

Figura 1515.Red Neuronal usada para identificar un sistema Fuente: Elaboración propia

3.3 Normalización de datos

$$X_{\text{normalized}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

Para la normalización de los datos, se necesitó de una ecuación, la cual X será el valor de kilovatios por meses menos Xmin que es el menor valor de los 12 meses, este resultado será dividido para la resta del valor máximo (Xmax) con el valor mínimo (Xmin).

NORMALIZACION DE DATOS 2017-2018 MES COMERCIAL INDUSTRIAL RESIDENCIAL NÚMERO DE CLIENTES ENERGIA mWh NÚMERO DE CLIENTES ENERGIA mWh NÚMERO DE CLIENTES ENERGIA mWh ene-17 13,159 9826,2873 154 8433,078 125,159 17388,1876 feb-17 13,097 9136,3069 155 15200,84 124,823 16965,1352 mar-17 13,08 8273,083 153 13007,255 124,737 16385,864 abr-17 13,077 9517,3889 154 16653,3 124,53 17943,5313 may-17 13,082 9374,0252 153 14833,878 124,753 18051,3232 jun-17 13,071 9594,2885 155 17237,401 125,112 17337,4368 jul-17 13,078 9090,4717 151 14041,487 125,9 15652,5259 ago-17 13,122 9163,0634 148 8811,14 126,255 15018,2595 sep-17 13,18 8808,3603 145 8438,063 126,734 15376,8186 oct-17 13,045 9087,9635 145 9330,049 127,001 15419,7374 nov-17 13,024 8993,0545 147 10426,488 127,583 15811,6591 dic-17 12,987 9049,3098 146 11100,026 127,392 15754,0741 ene-18 12,927 9472,0429 148 9135,793 127,024 17736,5433 feb-18 12,832 8969,8992 149 16352,454 126,653 16506,8684 mar-18 12,741 8416,0377 151 16880,181 125,995 17136,8828 abr-18 12,711 9401,6343 150 18101,224 126,065 18683,1247 may-18 13,082 9374,0252 153 14833,878 124,753 18051,3232 jun-18 12,666 9256,2627 149 20185,492 125,968 19018,3597 jul-18 12,651 7941,7485 152 20909,614 126,812 15388,032 ago-18 12,793 8468,6568 150 21426,055 127,707 15635,7063 sep-18 12,718 8353,9753 150 21812,372 127,558 16094,3134

Tabla 11.Normalización de datos-2017-2018 3.4

Ingreso datos a MATLAB

Para ingresar la información al software Matlab se debe declarar los valores iniciales con lo que se va a trabajar en este caso son los valores de consumo de energía a nivel residencial, comercial e industrial. A continuación, se explicará una tabla sencilla para representar una función XOR.

Tabla 22.Función XOR Patrón

X1

X2

XOR

X1

0

0

0

X2

0

1

1

X3

1

0

1

X4

1

1

0

Tomado de (Caicedo Bravo & López Sotelo, 2009)

Para ingresar los valores en el software las entradas en lugar de colocarlos como vector columna se la coloca como vector fila y en este caso XOR vendría a ser el target o valor objetivo también se lo coloca como otra matriz

egn

vector fila, es decir todas las entradas van en una sola matriz cada entrada como vector fila y las salidas en otra matriz como vector fila también.

Para la matriz de entrada de una función XOR, como se muestra en la Tabla 2, se va a tomar los valores de X1 y X2 y se los va a colocar dentro de un vector cuyo nombre del vector va a ser X, cada columna de la matriz va a estar separado por un punto y coma, que nos indica el cambio de columna. $x=[0\ 0\ 1\ 1;0\ 1\ 0\ 1]$ Para la matriz de salida de una función XOR, como se muestra en la Tabla 2, se va a tomar los valores de XOR y se los va a colocar dentro de un vector en este caso t.

$t=[0\ 1\ 1\ 0]$ El vector o matriz tiene que estar bajo ciertos parámetros los cuales son:

3.5

Creación de la red neuronal Las redes neuronales permiten la composición de diversas medidas las cuales permite resolver problemas de la forma en lo que realizaría un cerebro humano. Las redes neuronales en base a los parámetros de entrada nos permiten la obtención de salidas. Estas redes son conocidas como sistemas inteligentes dentro del área de la inteligencia artificial. 3.5.1 Newff

Es la que se encarga de crear un tipo de red Backpropagation, según su sintaxis deben ser especificados

con

los siguientes parámetros: newff (P, T, S) newff (P, T, S, TF, BTF, BLF, PF, IPF, OPF, DDF)

La descripción de newff (P, T, S) es la siguiente: • P - Matriz $R \times Q_1$ de Q_1 representativos de vectores de entrada de elementos. • T - Matriz $S_N \times Q_2$ de Q_2 vectores objetivo representativos del elemento S_N . • Si - Tamaños de capas ocultas $N-1$, S_1 a $S(N-1)$, por defecto = []. (El tamaño de capa de salida S_N se determina a partir de T.) Y por último devuelve una red backprop feed-forward de N capas. En cambio en newff (P, T, S, TF, BTF, BLF, PF, IPF, OPF, DDF) toma entradas opcionales, y su descripción es la siguiente:

- TFi - Función de transferencia de la capa i. El valor predeterminado es 'tansig' para capas ocultas, y 'purelin' para capa de salida. •

- BTF - Función de entrenamiento de red Backprop, por defecto = 'trainlm'. • BLF - Función de aprendizaje de peso / sesgo de Backprop, por defecto = 'learngdm'. • PF - Función de rendimiento, por defecto = 'mse'. •

- IPF: matriz de filas de funciones de procesamiento de entrada. El valor predeterminado es {'fixunknowns', 'remconstantrows', 'mapminmax'}. • OPF: matriz de filas de funciones de procesamiento de salida. El valor predeterminado es {'remconstantrows', 'mapminmax'}. • DDF - Función de división de datos, por defecto = 'dividerand'; y devuelve una red backprop feed-forward de N capas. Las funciones de transferencia TF {i} pueden ser cualquier transferencia diferenciable. Funciones como TANSIG, LOGSIG o PURELIN.

```
net=newff([0 1; 0 1],[2 1])
```

3.5.2 NEWP Crea un perceptrón. La función Newp en la versión R2010b NNET 7.0, está obsoleta mientras que en la versión R2010a NNET 6.0.4 es la más utilizada. Según su sintaxis deben ser especificados los siguientes parámetros.

```
net = newp (p, t, tf, lf)
```

La descripción de newp (p, t, tf, lf) es la siguiente: NET = newp (P, T, TF, LF) toma estas entradas, P - RxQ matriz de Q1 vectores de entrada representativos. T - Matriz SxQ de Q2 vectores objetivo representativos. TF - Función de transferencia, por defecto = 'hardlim'. LF - Función de aprendizaje, por defecto = 'learnp'. Devuelve un nuevo perceptrón. newp([0 1; 0 1],1)

3.6 Valores iniciales de la red Para inicializar la red es necesario adjuntar valores aleatorios iniciales de peso y umbrales, debido a que el peso tiene como función define la intensidad de conexión de las neuronas (p, t) mientras que el umbral, permite encender las neuronas iniciales de la red. Cada vez que se almacenan valores dentro de la red neuronal, a esto se denomina épocas, es decir que al ingresar 50 épocas por los 10 datos, se va obtener como resultado 500 datos. Las funciones de MATLAB, para inicializar la red tienen las siguientes descripciones: • net.inputWeights {1,1}. initFcn: Esta línea se refiere a la inicialización de los datos que son usado en la matriz. • net.biases {1,1}.initFcn: hace referencia a la longitud del vector. • net.trainParam.epochs=00; %: Hace referencia al máximo número de épocas. • net = init (net); %: Establece la Inicialización la red, este retorna los valores de la red. El siguiente fragmento de código hace referencia a los valores iniciales de la red: net.inputWeights {1,1}.initFcn='rands'; net.biases {1,1}.initFcn='rands'; net.trainParam.epochs=00; %Para aumentar el número de épocas net = init(net);%Inicializamos la red

3.7 Entrenamiento de la red Existen diferentes tipos de algoritmos de entrenamiento, para nuestro presente trabajo de investigación experimental vamos a enfocarnos en el algoritmo Backpropagation como se observa en la Figura 9.

Tabla 33.Funciones de entrenamiento con sus respectivos algoritmos Función de entrenamiento

Algoritmos Trainlm Levenberg-Marquardt Trainbr Regularización de Bayesiano Trainbfg Quasi-Newton BFGS Trainrp Resilient Backpropagation Trainscg Gradiente conjugado escala Traincgb Gradiente conjugado con Powell / Beale se reinicia Traincgf Gradiente conjugado Fletcher-Powell Traincgp Gradiente conjugado Polak-Ribière Trainoss Secante de un paso Traingdx Variable aprendizaje pendiente gradiente de velocidad. Traingdm Descenso de Gradiente con ímpetu. Traingd Descenso de Gradiente. Fuente: Elaboración propia 3.7.1 Trainlm Cambia los datos del peso, este algoritmo es considerado como el más ágil dentro del grupo de herramientas de MATLAB; este es sugerido como un algoritmo de supervisión, uno de los requerimientos para su uso es poseer buena capacidad de memoria ya que necesita más

aumento de memoria que los demás algoritmos. 3.7.2 Train Se centra en los algoritmos clásicos utilizados para entrenar redes neuronales, con base de códigos totalmente independiente, pero con la similitud que comparten con trainNetwork que ambas pertenecen a la misma caja de herramientas (Neural Network Toolbox). dentro de las ventajas de train, está la variedad de algoritmos.

```
net = train(net,p,t)
```

```
%Comprobamos que haya aprendido los patrones enseñados mediante la simulación. y =sim
(net, p) net = train(net,p,t) % La red tiene 1000 épocas net.IW{1,1} %Valor de los pesos net.b{1}
%Valor del bias
```

3.7.3 NNTOOL Es la interfaz gráfica de usuario (GUI) de Neural Network Toolbox, esta muestra la ventana administrador de red / datos que permite importar, crear, usar y exportar redes neuronales y datos, como se observa en la Figura 16.

Figura 16 GUI de nntool Fuente: MATLAB

En la figura 16, el input data, son los datos de entrada; target data, son los datos objetivos de la red neuronal; input delay states, es el estado de retardo de entrada; networks, son las redes resultantes; output data, es la salida de los datos; error data, es el margen de error de los datos; layer delay states, es el estado de retardo de capa.

3.8 Desempeño de la red neuronal

El desempeño de la red neuronal se centra en la resolución del problema en este caso es medir el consumo de energía, esta va a depender de las posibles soluciones que se obtengan, el desempeño depende del algoritmo que se emplee en la red neuronal, en este caso vamos a usar el algoritmo propagation. Por medio de este algoritmo se va medir el consumo de energía a nivel comercial, residencial e industrial. 3.9 Simulación de la red Sim.- Evalúa las salidas de red, es decir la representación gráfica de los resultados de la red neuronal, con estos resultados se puede llegar a una conclusión sobre el uso en potencia de los medidores comerciales, residenciales e industriales.

```
Y=sim (net, p) %Comprobamos que haya aprendido los patrones enseñados mediante la
simulación. y =sim (net, p) net = train (net,p,t) % La red tiene 1000 épocas net.IW{1,1} %Valor
de los pesos net.b{1} %Valor del bias y=sim (net, p) % Desempeño de la RNA para cada uno de
los patrones de la red entrenada %p1= [1; 1] %y1 = sim (net, p1)
```

```
%p2= [1; 0]
```

Dentro de los tipos de simulación de red neuronal, tenemos las siguientes: SimOut = sim (modelo, nombre, valor) simula el modelo especificado utilizando pares nombre-valor de parámetro.

SimOut = sim (model, ParameterStruct) simula el modelo especificado utilizando los valores de los parámetros especificados en la estructura ParameterStruct.

SimOut = sim (modelo, ConfigSet) simula el modelo especificado utilizando los ajustes de configuración especificados en el conjunto de configuración del modelo ConfigSet.

SimOut = sim (modelo) simula el modelo especificado utilizando los parámetros de configuración del modelo existente y devuelve el resultado como un objeto. Para devolver los resultados de la simulación utilizando el formato de salida única (objeto de simulación), seleccione Salida de simulación única en el panel Importar / Exportar datos del cuadro de diálogo Parámetros de configuración. Esta selección anula el formato de conjunto de datos utilizado para el registro de señales. Para devolver los resultados de la simulación utilizando el formato compatible con versiones anteriores (vector de tiempo), consulte Sintaxis compatible con versiones anteriores.

CAPÍTULO 4 DESARROLLO DEL TEMA En base la siguiente información recolectada de la empresa CNEL-Milagro se procedió a normalizarlos para luego ingresarlo a la herramienta Matlab, y por último se realizó la simulación de las redes neuronales. Tabla 44.Datos Normalizados CNEL-Milagro CONSUMO DE ENERGIA mwh 2017-2018

MES NÚMERO DE CLIENTES COMERCIAL COMERCIAL NÚMERO DE CLIENTES INDUSTRIAL INDUSTRIAL NÚMERO CLIENTES RESIDENCIAL RESIDENCIAL

0: urkund onofre_campoverde.docx

100%

ene-17 0,89119171 1 0,9 0 0,206026859 0,781364434 feb-17 0,569948187 0,555769708 1
0,768686247 0,095971176 0,641884211 mar-17 0,481865285 0 0,8 0,519537618 0,067802162
0,450898707 abr-17 0,466321244 0,801121849 0,9 0,933657477 0 0,96446105 may-17
0,492227979 0,708819954 0,8 0,727006494 0,073042909 1 jun-17 0,435233161 0,850632142 1
1 0,190632165 0,764631913 jul-17 0,471502591 0,526259617 0,6 0,637006275 0,448738945
0,209117402 ago-17 0,699481865 0,572996353 0,3 0,042940496 0,565018015 0 sep-17 1
0,344627748 0 0,000566199 0,721912873 0,118216805 oct-17 0,300518135 0,524644762 0
0,101878475 0,809367835 0,132367118 nov-17 0,191709845 0,463539471 0,2 0,226412638 1
0,26158356 dic-17 0 0,499758338 0,1 0,302913466 0,937438585 0,242597806

ene-18 0,64037123 1 0 0 0,768788084 0,646914409 feb-18 0,419953596 0,671864643 0,2
0,569290895 0,643195667 0,308191572 mar-18 0,208816705 0,309933304 0,6 0,610920975
0,420446852 0,481733591 abr-18 0,139211137 0,95399016 0,4 0,707243729 0,444143534
0,907657097 may-18 1 0,935948468 1 0,449497061 0 0,733622808 jun-18 0,034802784
0,858994322 0,2 0,871662536 0,411306703 1 jul-18 0 0 0,8 0,928785361 0,697020988 0 ago-18
0,329466357 0,344318257 0,4 0,969525138 1 0,068223676 sep-18 0,155452436 0,269377448
0,4 1 0,949559919 0,194550316

4.1 Pasos para la simulación desarrollada en el software MATLABI

Figura 17.Creación de la red neuronal-Cliente Comerciales Fuente: Elaboración propia

Se crea la red multicapa para el sector comercial usando el algoritmo de retro propagación, luego se procede a ingresar las entradas que son los datos de los clientes del sector

comercial, mientras que las salidas son el consumo del mismo sector. Utilizamos las funciones `trainlm` (es un algoritmo que permite la actualización los valores de peso, y sesgo), `learnngdm` (se refiere al descenso del gradiente que consta de impulso de peso y funciones de aprendizaje), ingresamos 4 capas ocultas, cada capa con 10 neuronas y para la función de activación `tansig` aplicamos la sigmoideal que se refiere a la transferencia neuronal, luego se crea la red neuronal, como se observa en la Figura 17.

Se realiza el mismo procedimiento anterior para los clientes del sector industrial y residenciales, es decir con el mismo número de capas y funciones, como se observa en la Figura 18.

Figura 18. Creación de la red neuronal Clientes Industriales y Residenciales. Fuente: Elaboración propia

Una vez creadas las tres redes para cada sector una red diferente, damos doble clic sobre cada una de las redes, en este caso hemos tomado de ejemplo los datos de la red comercial, hacemos doble clic donde nos muestra la estructura de la red, por medio de esto se observa la red neuronal con 4 capas y 10 neuronas, una entrada, una salida como se observa en la Figura 19. Figura 1719. Estructura de la Red Neuronal Fuente: Elaboración Propia

Posteriormente damos clic en la función `train` donde se realizó el entrenamiento de la red, para esto se utiliza el número de clientes comerciales normalizados como entrada y la salida es el consumo comercial, se le asignó un nombre para que se guarden los resultados en la red comercial salida y los errores entre el valor verdadero y el valor estimado, como se observa en la Figura 20.

Figura 20. Entrenamiento de la Red Neuronal Fuente: Elaboración propia

Damos clic en la opción `Training Parameters`, donde se asignó el número de épocas en este caso se escogieron 1000 épocas, se ingresó el coeficiente de aprendizaje que debe estar entre el valor 0,1, como se observa en la Figura 21.

Figura 21. Parámetros de Entrenamiento Fuente: Elaboración Propia

Luego damos clic en `train network`, nos muestra una ventana donde se entrena la red una vez entrenado nos muestra 4 gráficas que tienen que ver con los coeficientes R de entrenamiento, validación y test, como se observa en la Figura 22.

Figura 22. Entrenamiento de la Red Fuente: Elaboración Propia

Figura 23. Simulación de la red neuronal Fuente: Elaboración Propia Una vez que se ha conseguido que el coeficiente R tenga valores cercanos a 1 se procede a la simulación con los datos para saber que ha sucedido en la red, como se observa en la Figura 23. En base a los resultados de la simulación nos muestra los siguientes valores. Como se observa en la Figura 24.

Figura 1824. Resultados cuantitativo de la Red Neuronal Fuente: Elaboración Propia

Seleccionamos un dato al azar que no se encuentra en nuestra tabla normalizada por medio de la siguiente fórmula.

$$x = X \cdot X_{MAX} - X_{MIN} + X_{MIN}$$

4.1.2 Pasos para el obtener los resultados en Kilovatios • Mediante esta fórmula determinamos el número de personas $x = X \cdot X_{MAX} - X_{MIN} + X_{MIN}$ $x = 0,809713,159 - 12,651 + 12,651$ $x = 13,062$ • Mediante esta fórmula determinamos el número de kilovatios $x = X \cdot X_{MAX} - X_{MIN} + X_{MIN}$ $x = 0,69229826,2873 - 7941,7485 + 7941,7485$ $x = 9246,226$ kw Una vez obtenidos los resultados que nos da la herramienta del software Matlab se aplica un proceso inverso en base a la fórmula de normalización, realizando un despeje para que nos muestre el valor normal en kilovatios, en base a la simulación del consumo que se espera tener, se realiza las tomas de decisiones en la empresa CNEL-Milagro.

CAPÍTULO 5

CONCLUSIONES

Mediante el presente trabajo de investigación experimental se logró cumplir con los objetivos establecidos, es decir se realizó la simulación en el software Matlab en base a los datos proporcionados por la empresa CNEL-Milagro, llegando a las siguientes conclusiones: • Las redes neuronales son parte de la inteligencia artificial capaces de aprender y mejorar su funcionamiento y realizar predicciones en secuencia de tiempo. • Por medio del software Matlab se logró conocer la herramienta y todo lo necesario para poder utilizarla, es decir los valores de peso, épocas, umbrales, número de neuronas y coeficiente de aprendizaje. • El software Matlab proporciona una interfaz gráfica nntool la cual permitió realizar el ingreso de datos y visualizar como la red neuronal estaba en su proceso de entrenamiento para adquirir conocimiento. • Mediante la herramienta Matlab se obtuvieron resultados cuantitativos a los cuales se les tuvo que aplicar un proceso inverso por medio de la fórmula de normalización para obtener resultados en kilovatios, según los resultados obtenidos se puede realizar la toma de decisiones respecto al consumo de energía en los diferentes grupos los cuales son residencial, comercial e industrial. • Como se pudo observar en el desarrollo del tema solo se consideró a un grupo específico de consumo, en este caso fue el sector comercial logrando realizar la simulación de los datos para así llegar a un valor específico expresado en kilovatios.

Comment11 Punto o coma para enlazar la frase Comment12 Si no va a mencionar cuales son, mejor no pong esto Comment13 No hay un enlace coherente con lo que menciona antes de la coma Comment21 Mencione para que va a usar la red neuronal Comment22 Comment93 No está clara la fórmula 33

1

[Metadata removed]

Hit and source - focused comparison, Side by Side:

Left side: As student entered the text in the submitted document.

Right side: As the text appears in the source.

Instances from: urkund onofre_campoverde.docx

0: urkund onofre_campoverde.docx 84%

ya sea pasado, presente o futuro. El análisis predictivo se centra en la identificación

1: urkund onofre_campoverde.docx 86%

el mes de consumo se lo conoce como conjunto de entrenamiento. Se ingresarán datos de dos años, a cada conjunto de valores ingresados

2: urkund onofre_campoverde.docx 92%

y se procede a utilizar la siguiente fórmula que consiste en la división del número de consumo por mes, entre el total

0: urkund onofre_campoverde.docx 84%

ya sea en el pasado, presente o futuro. El análisis predictivo se basa en la identificación

1: urkund onofre_campoverde.docx 86%

el mes de consumo, a esto se lo conoce como conjunto de entrenamiento.

Se ingresaran datos de dos años, a cada conjunto de valores ingresados

2: urkund onofre_campoverde.docx 92%

y se procede a utilizar la siguiente formula que consiste en la división del número de consumo por mes entre el total

3: urkund onofre_campoverde.docx 100%

vector fila, es decir todas la entradas van en una sola matriz cada entrada como vector fila y las salidas en otra matriz como vector fila también.

4: urkund onofre_campoverde.docx 100%

ene-17 0,89119171 1 0,9 0 0,206026859 0,781364434 feb-17
 0,569948187 0,555769708 1 0,768686247 0,095971176
 0,641884211 mar-17 0,481865285 0 0,8 0,519537618
 0,067802162 0,450898707 abr-17 0,466321244 0,801121849 0,9
 0,933657477 0 0,96446105 may-17 0,492227979 0,708819954 0,8
 0,727006494 0,073042909 1 jun-17 0,435233161 0,850632142 1 1
 0,190632165 0,764631913 jul-17 0,471502591 0,526259617 0,6
 0,637006275 0,448738945 0,209117402 ago-17 0,699481865
 0,572996353 0,3 0,042940496 0,565018015 0 sep-17 1
 0,344627748 0 0,000566199 0,721912873 0,118216805 oct-17
 0,300518135 0,524644762 0 0,101878475 0,809367835
 0,132367118 nov-17 0,191709845 0,463539471 0,2 0,226412638
 1 0,26158356 dic-17 0 0,499758338 0,1 0,302913466 0,937438585
 0,242597806

3: urkund onofre_campoverde.docx 100%

vector fila, es decir todas la entradas van en una sola matriz cada entrada como vector fila y las salidas en otra matriz como vector fila también.

4: urkund onofre_campoverde.docx 100%

ene-17 13,159 9826,2873 154 8433,078 125,159 17388,1876
 feb-17 13,097 9136,3069 155 15200,84 124,823 16965,1352
 mar-17 13,08 8273,083 153 13007,255 124,737 16385,864 abr-17
 13,077 9517,3889 154 16653,3 124,53 17943,5313 may-17 13,082
 9374,0252 153 14833,878 124,753 18051,3232 jun-17 13,071
 9594,2885 155 17237,401 125,112 17337,4368 jul-17 13,078
 9090,4717 151 14041,487 125,9 15652,5259 ago-17 13,122
 9163,0634 148 8811,14 126,255 15018,2595 sep-17 13,18
 8808,3603 145 8438,063 126,734 15376,8186 oct-17 13,045
 9087,9635 145 9330,049 127,001 15419,7374 nov-17 13,024
 8993,0545 147 10426,488 127,583 15811,6591 dic-17 12,987
 9049,3098 146 11100,026 127,392 15754,0741