REDES NEURONALES ARTIFICIALES APLICADAS AL RECONOCIMIENTO DE PATRONES

WILMER RIVAS ASANZA / BERTHA MAZÓN OLIVO

Editorial UTMACH

REDES 2017

COLECCIÓN EDITORIAL

Redes neuronales artificiales aplicadas al reconocimiento de patrones

Wilmer Rivas Asanza
Bertha Mazón Olivo

Coordinadores

Primera edición en español, 2018

Este texto ha sido sometido a un proceso de evaluación por pares externos con base en la normativa editorial de la UTMACH

Ediciones UTMACH

Gestión de proyectos editoriales universitarios

199 pag; 22X19cm - (Colección REDES 2017)

Título: Redes neuronales artificiales aplicadas al reconocimiento de patrones. / Wilmer Rivas Asanza / Bertha Mazón Olivo (Coordinadores)

ISBN: 978-9942-24-100-9

Publicación digital

Título del libro: Redes neuronales artificiales aplicadas al reconocimiento de patrones.

ISBN: 978-9942-24-100-9

Comentarios y sugerencias: editorial@utmachala.edu.ec

Diseño de portada: MZ Diseño Editorial

Diagramación: MZ Diseño Editorial

Diseño y comunicación digital: Jorge Maza Córdova, Ms.

© Editorial UTMACH, 2018

© Wilmer Rivas / Bertha Mazón, por la coordinación

D.R. © UNIVERSIDAD TÉCNICA DE MACHALA, 2018

Km. 5 1/2 Vía Machala Pasaje

www.utmachala.edu.ec

Machala - Ecuador

Advertencia: “Se prohíbe la reproducción, el registro o la transmisión parcial o total de esta obra por cualquier sistema de recuperación de información, sea mecánico, fotoquímico, electrónico, magnético, electro-óptico, por fotocopia o cualquier otro, existente o por existir, sin el permiso previo por escrito del titular de los derechos correspondientes”.

César Quezada Abad, Ph.D
Rector

Amarilis Borja Herrera, Ph.D
Vicerrectora Académica

Jhonny Pérez Rodríguez, Ph.D
Vicerrector Administrativo

COORDINACIÓN EDITORIAL

Tomás Fontaines-Ruiz, Ph.D
Director de investigación

Karina Lozano Zambrano, Ing.
Jefe Editor

Elida Rivero Rodríguez, Ph.D
Roberto Aguirre Fernández, Ph.D
Eduardo Tusa Jumbo, Msc.
Irán Rodríguez Delgado, Ms.
Sandy Soto Armijos, M.Sc.
Raquel Tinóco Egas, Msc.
Gissela León García, Mgs.
Sixto Chiliquinga Villacis, Mgs.
Consejo Editorial

Jorge Maza Córdova, Ms.
Fernanda Tusa Jumbo, Ph.D
Karla Ibañez Bustos, Ing.
Comisión de apoyo editorial

Índice

Capítulo I

Generalidades de las redes neuronales artificiales 11

Wilmer Rivas Asanza; Bertha Mazón Olivo; Edwin Mejía Peñafiel

Capítulo II

Tecnologías utilizadas en el reconocimiento de patrones y clasificación de Imágenes 36

Wilmer Rivas Asanza; Bertha Mazón Olivo; Eduardo Tusa

Capítulo III

Reconocimiento de patrones en imágenes 61

Wilmer Rivas Asanza; Bertha Mazón Olivo; Eduardo Tusa

Capítulo IV

Matemáticas aplicadas en la clasificación de patrones 127

Maritza Alexandra Pinta; Mónica Loaiza Loayza; Freddy Jumbo Castillo

Capítulo V

Reconocimiento de patrones en imágenes médicas. Caso de estudio Mamografías.....176

Eduardo Tusa; María Perez

Dedicatoria

A nuestra familia por su apoyo incondicional y comprensión.

A los estudiantes de la Universidad Técnica de Machala, Carrera de Ingeniería de Sistemas por su aporte valioso

Introducción

Las Redes neuronales artificiales (RNA), es una técnica de la Inteligencia Artificial, debido a que intenta desarrollar soluciones informáticas emulando el comportamiento del cerebro humano. La capacidad del cerebro humano de pensar, recordar y resolver problemas ha inspirado a muchos científicos intentar o procurar modelar en el ordenador el funcionamiento del cerebro humano.

En la actualidad son ampliamente usadas en diferentes campos, por ejemplo reconocimiento de imágenes a través de patrones. En este sentido, también los especialistas, expertos, etc., han desarrollado tecnología que permite modelar los diferentes tipos de redes neuronales. Es por ello, que este texto propone abarcar el estudio de las diferentes herramientas tecnológicas, ejemplos prácticos, la relación que existe entre las RNA y las matemáticas y casos prácticos desarrollados por los autores en el ámbito de la medicina y la agronomía.

02 Capítulo

Tecnologías utilizadas en el reconocimiento de patrones y clasificación de imágenes

Wilmer Rivas Asanza; Bertha Mazón Olivo; Eduardo Tusa

Este capítulo tiene como objetivo principal presentar algunas herramientas tecnológicas que permiten modelar las redes neuronales artificiales, se muestra el uso de la herramienta a través de capturas de pantalla, de tal manera que se facilite al lector la utilización de las mismas al emprender en proyectos que usen la técnica de redes neuronales artificiales.

Wilmer Rivas Asanza: Ingeniero en Sistemas, Diplomado en Auditoría Informática, Magíster en Docencia y Gerencia en Educación Superior, Magíster en Gestión Estratégica en Tecnologías de la Información además de cursar un programa de Doctorado en TIC en España. Soy docente titular en la UTMACH. A la fecha cuento con varios publicaciones indexadas y libros.

Bertha Mazón Olivo: Ingeniera en Sistemas y Magíster en Informática Aplicada. Cuenta con 15 años de experiencia docente y profesional en el campo de la Informática. Actualmente es Profesor Titular en la UTMACH y curso el programa de doctorado en TIC en España. A la fecha cuenta con varias publicaciones indexadas y libros.

Eduardo Tusa: Ingeniero Electrónico (Magna Cum Laude) con una Subespecialización en Matemáticas de la Universidad San Francisco de Quito. Su cuarto año de formación de pregrado fue realizado en USA. Máster en Visión, Imagen y Robótica (con distinción) de la Universidad de Borgoña, la Universidad de Girona y la Universidad Heriot-Watt. Es docente de la Unidad Académica de Ingeniería Civil de la UTMACH.

Joone - Java

Java tiene su propia estructura y sintaxis, se basa en la Programación Orientada a Objetos (POO), los atributos y funciones son puestos dentro de una clase (encapsulados) usada para generar objetos de un tipo en particular. (Adam, 2015)

Java es un lenguaje cross-platform, lo que permite la ejecución de cualquier aplicación creada sea ejecutada en diferentes sistemas operativos tanto Windows, Linux, MacOSX, esta independencia es factor clave.

Existe una extensa comunidad de programadores alrededor del mundo que contribuyen constantemente al desarrollo de librerías y frameworks open source para la inteligencia artificial. JOONE es un framework dedicado al reconocimiento de patrones se respalda en algoritmos matemáticos complejos, cumple un papel importante en el desarrollo de aplicaciones que ayudan a la predicción y/o detección de patrones a partir de imágenes, por ejemplo en la detección de rostros, firmas, placas de vehículos y porque no también en el ámbito de la salud por ejemplo la detección a tiempo de un cáncer de mama.

JGAP es otro framework para el campo de la inteligencia artificial, incorpora funciones de algoritmos genéticos que son usados para aplicar los principios de la evolución a diferentes áreas de la vida real, resolver problemas de búsquedas (ruta más corta) y optimización (mejorar algo hasta su punto máximo)

Joone (Marrone, 2007) es un framework propio de Java para crear y ejecutar aplicaciones de Inteligencia Artificial basadas en redes neuronales.

Joone puede ser considerado más que un marco de redes neuronales para un entorno completo de desarrollo integrado. A diferencia de sus contrapartes comerciales, tiene un fuerte enfoque en el desarrollo basado en el código de las redes neuronales en lugar de la construcción visual.

Aunque en teoría Joone se puede utilizar para construir una gama más amplia de sistemas adaptativos (incluyendo aquellos con elementos no-adaptativas), su atención se centra en las redes neuronales basado en backpropagation (Propagación hacia atrás).

- Las aplicaciones de Joone se pueden construir en una máquina local, estar entrenado en un entorno distribuido y ejecutado en cualquier dispositivo
- Joone consiste en una arquitectura basada en componentes combinables que se pueden extender para construir nuevos algoritmos de aprendizaje y redes neuronales arquitecturas.
- Los componentes son módulos de código de plug-in que se ligan para producir un flujo de información. Los nuevos componentes se pueden añadir y volver a utilizar. Más allá de la simulación, Joone también tiene en cierta medida las capacidades de despliegue multi-plataforma.
- Joone tiene un editor de interfaz gráfica de usuario para crear de forma gráfica y probar cualquier red neuronal, y un entorno de formación distribuida que permite a las redes neurales para ser capacitados en múltiples máquinas remotas.

Dentro de las aplicaciones de Joone con enfoque hacia dar soluciones óptimas a problemas relacionados con la Inteligencia Artificial basados en las redes neuronales tenemos los siguientes:

Red neuronal difusa para un planificador de entornos GRID: (Garmendia & Santos, pág. 8) “Los entornos Grid son sistemas distribuidos muy poco acoplados que utilizan recursos conectados por medio de redes para resolver problemas computacionales de alto rendimiento. A diferencia de sistemas clúster tradicionales, pueden compartir recursos de computación heterogéneos y de almacenamiento.”

Predicción de Series de Tiempo: Joone es aplicado en el área financiera para el mercado de valores.

Empezando con Joone

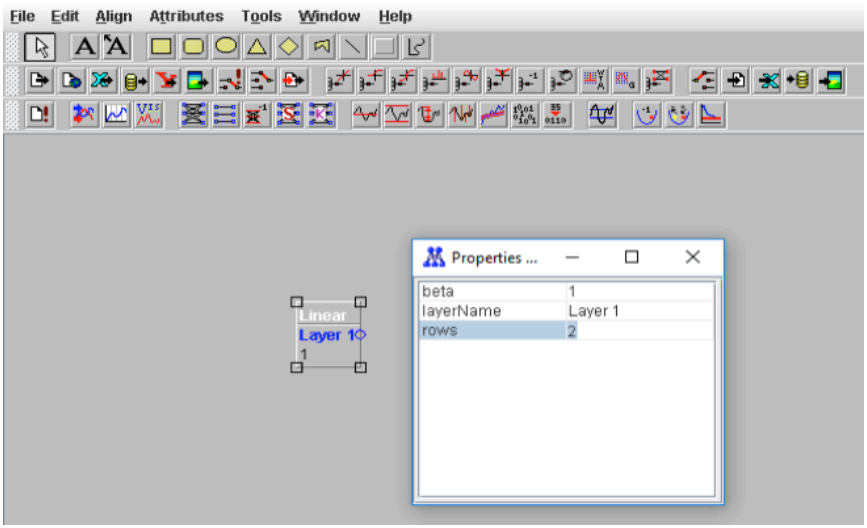
Requerimientos: Para que pueda usar Joone en su ordenador necesita: Java VM o también se lo llama (Java Runtime Environment).

Instalación y configuración: Se debe iniciar el ejecutable JooneEditor2.0.0RC1-windows.exe en sistemas operativos Windows y JooneEditor2.0.0RC1-linux.sh para sistemas operativos Linux, inmediatamente aparece el asistente de instalación indicando los pasos a seguir para continuar.

Creación de redes neuronales con Joone

- Se debe crear un archivo txt y lo guardamos en una carpeta, el nombre de la carpeta no importa solo se tiene que saber dónde se lo va a guardar el archivo.txt en este caso se lo llama al archivo rna.txt
- Se procede abrir JooneEditor.
- Ahora se procede a dar click en el icono que dice “new linear layer” para luego colocar sobre el espacio de trabajo que nos ofrece JooneEditor.
- Luego se deberá dar click derecho sobre él y damos click en propiedades en y a continuación escogemos rows que hace referencia a las filas en la cual se establece el valor de 2.

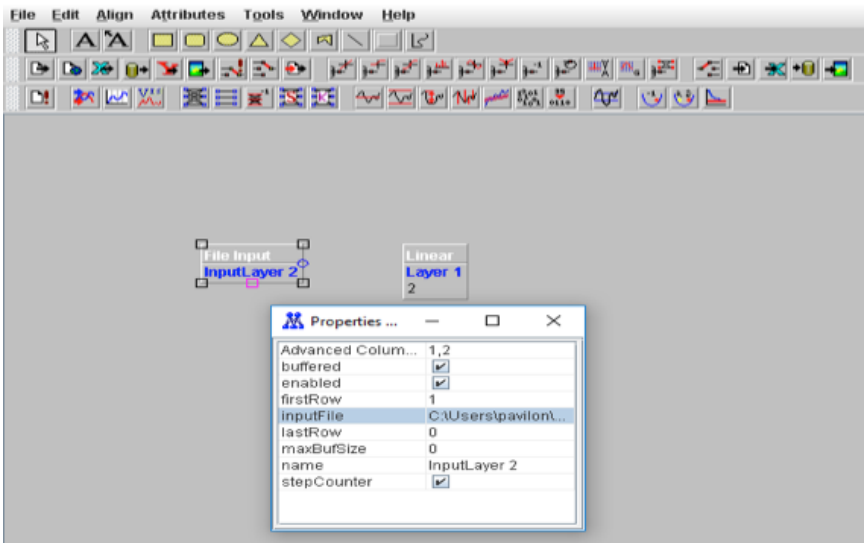
Ilustración 1: Configuración del “new linear layer”



Fuente: propia

- Ahora se procede a crear un “new file input layer” en elemento sirve una herremienta para hace referencia a los datos de entrada.
- Se procede a dar click sobre él y en propiedades, en advance column se escribe 1,2, en input file se da click sobre el icono que se aparece y se muestra una ventana.
- Escogemos el archivo que se creó con el nombre de rna.txt

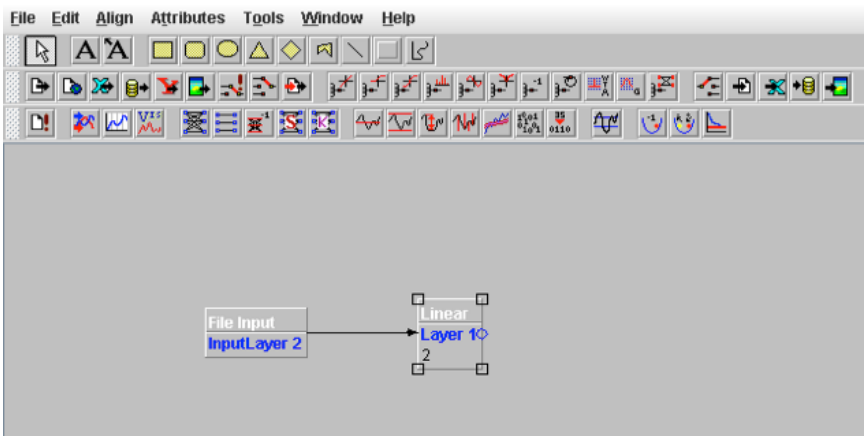
Ilustración 2: Configuración del “new file input layer.



Fuente: propia

- Se conecta el new input file layer con el new linear layer.

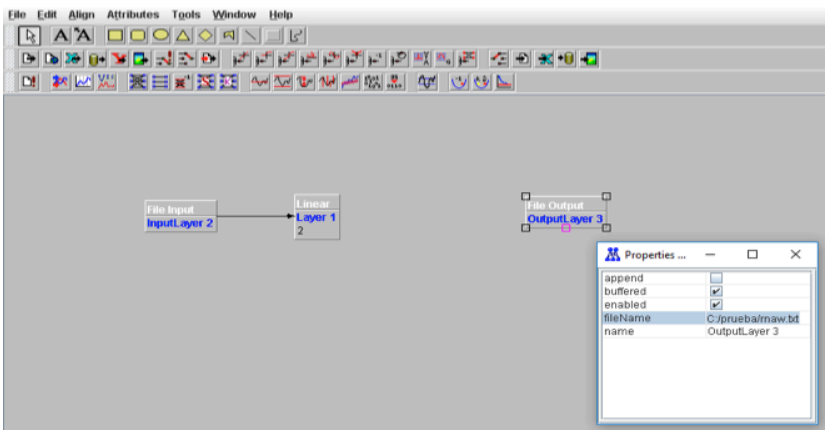
Ilustración 3: Relación entre el “new linear layer” y “new file input layer”



Fuente: propia

- Ahora se procede a crear un “new output layer” este elemento servirá para poder mostrar los datos de salida.
- Procedemos a dar click derecho sobre él y en propiedades se escoge la opción finename en la cual se escribe la dirección donde se quiere que se cree el archivo de salida en este caso se procede a escoger la Unidad C, en una carpeta que contendrá un archivo llamado “rnaw.txt”.

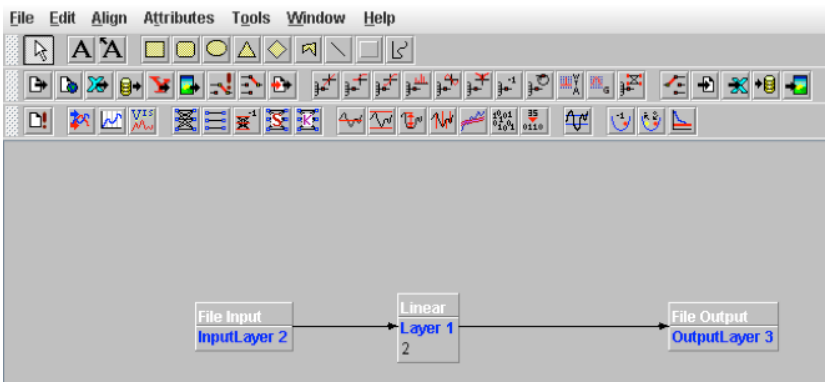
Ilustración 4: Configuración del “new output layer”



Fuente: propia

- Luego procedemos a realizar las relaciones correspondientes.

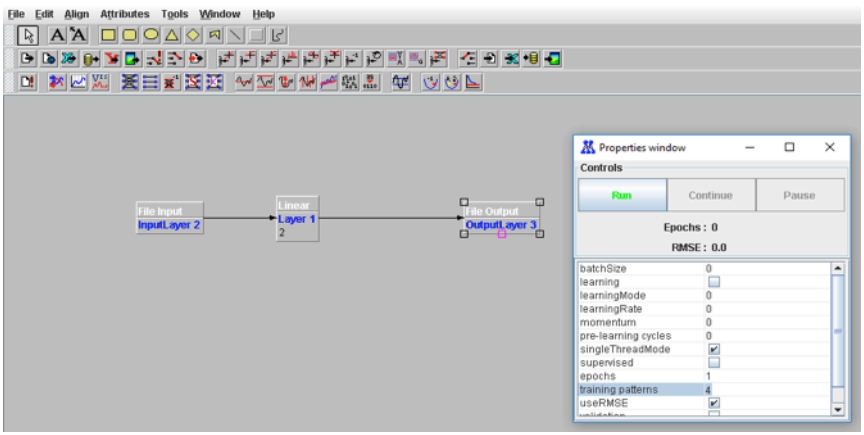
Ilustración 5: Relación entre “new linear layer”, “new file input layer” y “new output layer”



Fuente: propia

- Seguido se procede a dar click en Tools que se encuentra en la barra de herramientas del software y se escoge la opción Control Panel.
- Aparcera una ventana en epochs igual a 1 y en training patterns igual a 4.
- Se procede a ejecutar la red neuronal en la cual se crea un archivo resultante con el nombre rnaw.txt anteriormente establecido en el new output file.

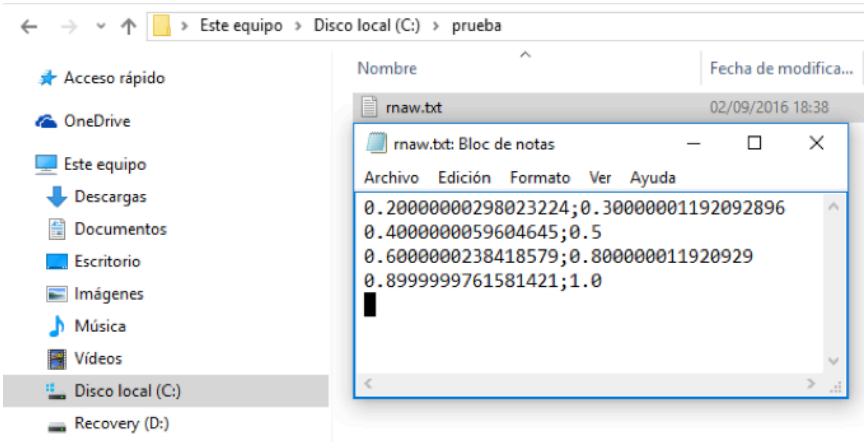
Ilustración 6: Ejecución de la Red Neuronal



Fuente: (Autor)

- No ubicamos en la Unidad C y procedemos a ingresar a la carpeta donde se creó el primer archivo en la cual se tendrá otro archivo resultante.
- Como se pude ver en el archivo se muestran los datos aprendidos por la red neuronal.

Ilustración 7: Datos aprendidos por la Red Neuronal



Fuente: (Autor)

Neuro Estudio

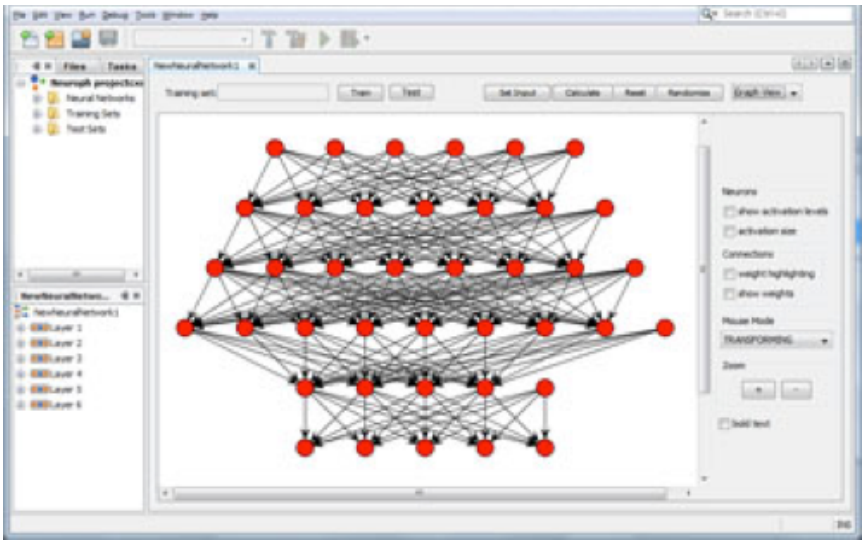
Neuroph Studio es un entorno de interfaz gráfica de usuario que utiliza todas las funcionalidades del lenguaje de programación Java necesarias para desarrollo de redes neuronales, éste viene incluido en la plataforma NetBeans que también utiliza las aplicaciones Java Swing para trabajar. Además, los desarrolladores pueden crear, probar e implementar componentes de redes neuronales desde el mismo IDE NetBeans utilizando Java, es decir, sin necesidad de hacerlo como antes que el desarrollador debía programar redes neuronales y también programar lenguaje Java. Algunas de las funcionalidades que se pueden resolver de redes neuronales en la herramienta Neuroph Studio es reconocimiento, predicción, control y clasificación de problemas mal definidos. (Sevarac, Oracle, 2011)

Proyectos donde se utilice neuroph

Neuroph Studio es ideal también para la simulación de experimentos basándose en el marco de red neuronal de Java, extrayendo características importantes como energía, contraste, homogeneidad y correlación. Es una herramienta de código abierto y bajo Licencia Apache 2.0.

Neuroph es una herramienta ideal para ser utilizada por profesionales y principiantes que deseen llevar a la práctica la teoría de redes neuronales, ya que la herramienta se encuentra bien documentada, tiene un entorno de interfaz amigable y fácil de usar con flexibilidad de uso.

Ilustración 8. Área de trabajo de Neuroph Studio



Fuente: (Neuroph, S/A)

Empezando con Neuroph Studio

Requerimientos: Para que pueda usar Neuroph Studio en su ordenador necesita: Java VM 1.7.

Instalación y configuración: Se debe iniciar el ejecutable neurophstudio - windows.exe en sistemas operativos Windows y neurophstudio-linux.sh para sistemas operativos Linux, inmediatamente aparece el asistente de instalación indicando los pasos a seguir para continuar.

Además, se debe añadir referencias a la biblioteca Neuroph -core - xx.jar a su proyecto de Java (y otras independencias como Neuroph - imgrec - 2.7.jar en caso de utilizar reconocimientos de imágenes).

Creación de redes neuronales con Neuroph Studio

Para la creación de redes neuronales en Neuroph Studio se deben seguir los siguientes pasos:

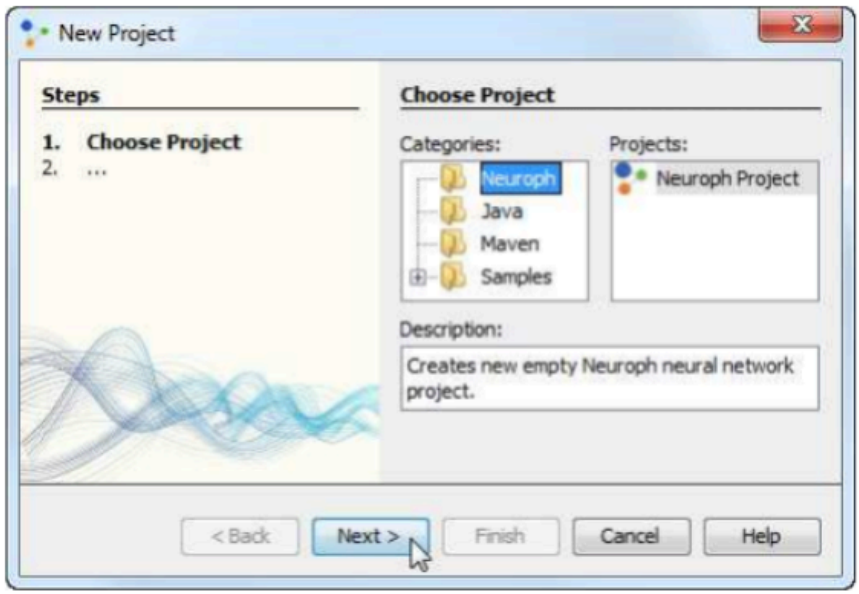
Paso 1: Crear un nuevo proyecto Neuroph.

Ilustración 9: Crear nuevo proyecto Neuroph

Fuente: propia

Seleccionar un proyecto Neuroph y clic en siguiente.

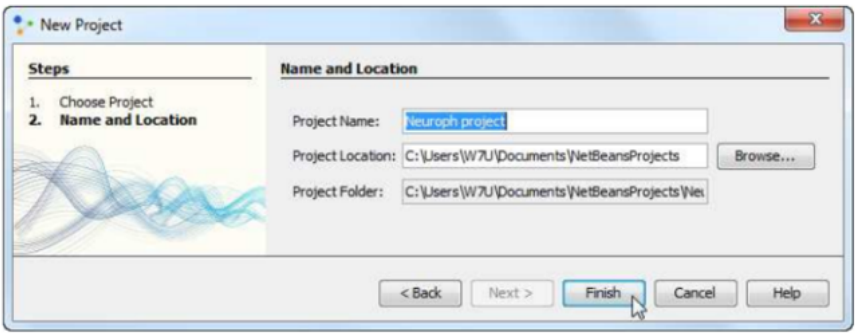
Ilustración 10: Selección de proyecto Neuroph



Fuente: propia

Ingresar el nombre y la ubicación del proyecto, clic en Finalizar

Ilustración 11: Ingresar nombre y ubicación del proyecto



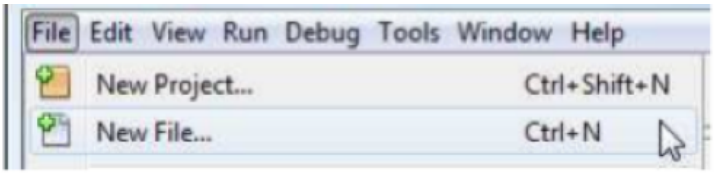
Fuente: propia

El proyecto se encuentra creado, ahora se debe crear la red neuronal.

Paso 2: Creación de la red Perceptrón.

Clic en Archivo > Nuevo Archivo

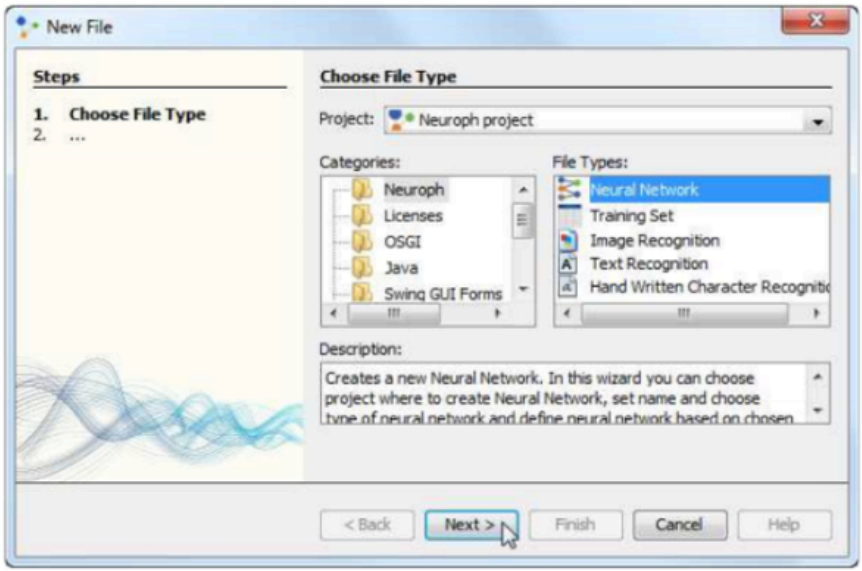
Ilustración 12: Crear Nuevo Archivo



Fuente: propia

Seleccionar proyecto deseado desde el menú desplegable Proyecto, Neuroph como categoría, redes neuronales de tipo de archivo y haga clic en Siguiente

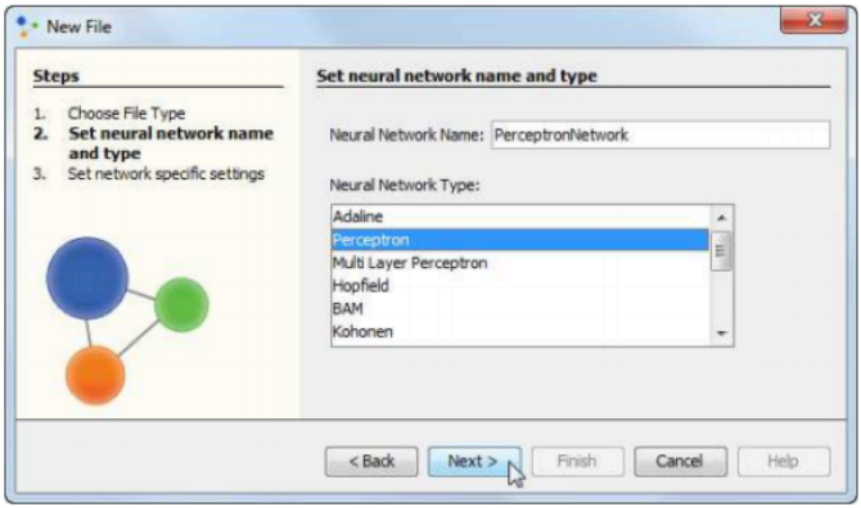
Ilustración 13: Seleccionar proyecto



Fuente: propia

Ingresar el nombre de la red, seleccione el tipo de red Perceptrón, y clic en Siguiente.

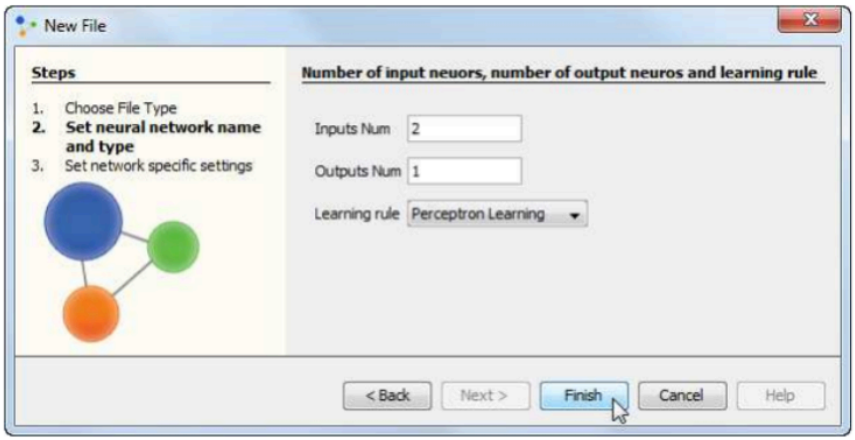
Ilustración 14: Selección de tipo de red



Fuente: propia

En un nuevo cuadro de dialogo se debe introducir el número de neuronas en entrada y salida. Elegir el aprendizaje Perceptrón y clic en Finalizar.

Ilustración 15: Ingreso número de neuronas



Fuente: propia

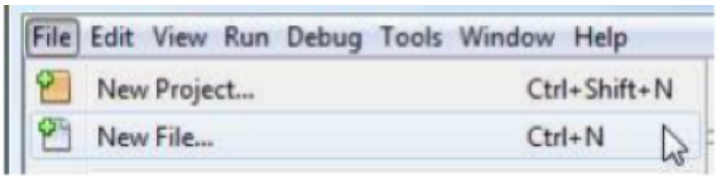
Se ha creado la red neuronal perceptrón con dos neuronas en la entrada, y uno en la capa de salida.

Ilustración 16: creación de neuronas

Fuente: propia

Para crear el conjunto de entrenamiento, haga clic en Archivo > Nuevo Archivo, para abrir asistente conjunto de entrenamiento.

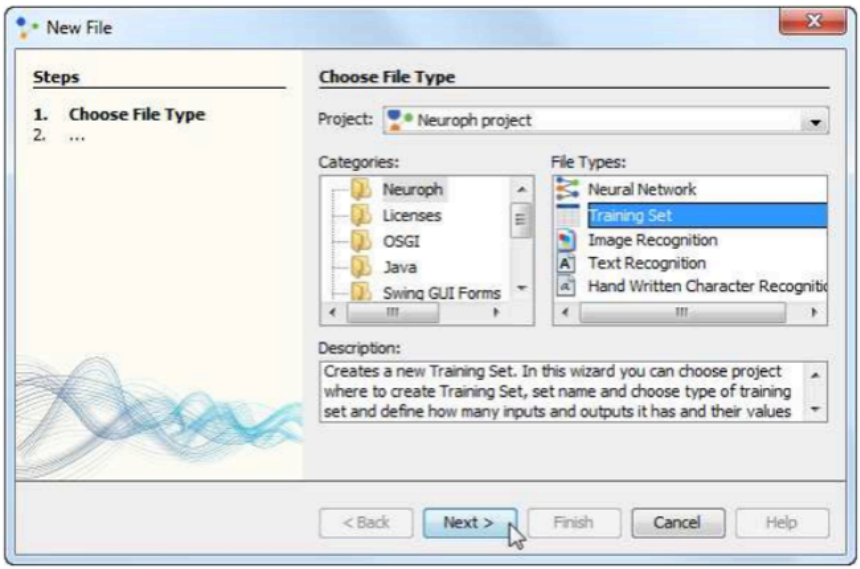
Ilustración 17: Nuevo Archivo



Fuente: propia

Elija un tipo de archivo conjunto de entrenamiento, haga clic en Siguiente.

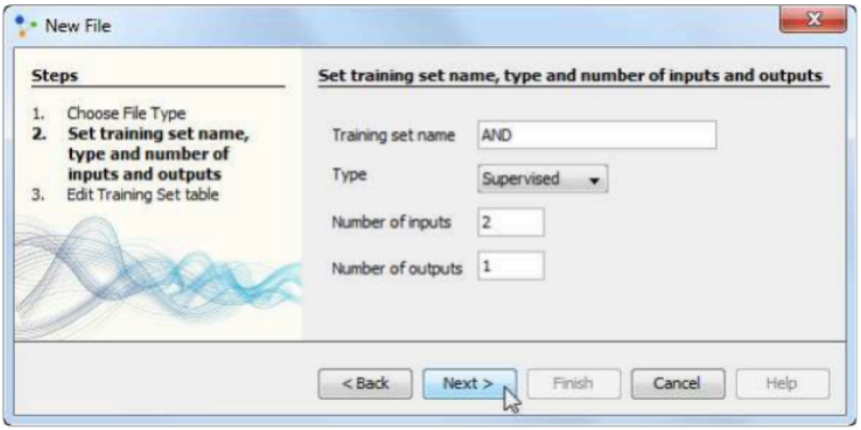
Ilustración 18. Seleccionar archivo para el entrenamiento.



Fuente: propia

Ingrese el nombre del conjunto, el número de entradas y salidas como se muestra en la imagen de abajo y haga clic en el botón Crear

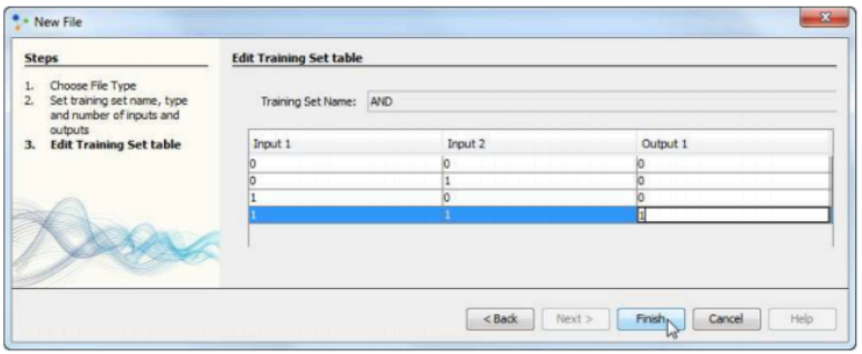
Ilustración 19. Asignar training, tipo, y número de entradas y salidas.



Fuente: propia

A continuación, cree conjunto de entrenamiento mediante la introducción de elementos de formación como valores de entrada y de salida deseadas de las neuronas en la entrada y la capa de salida. Utilice el botón Añadir fila para añadir nuevos elementos y haga clic en el botón OK cuando haya terminado.

Ilustración 20. Edición de datos para entrenamiento

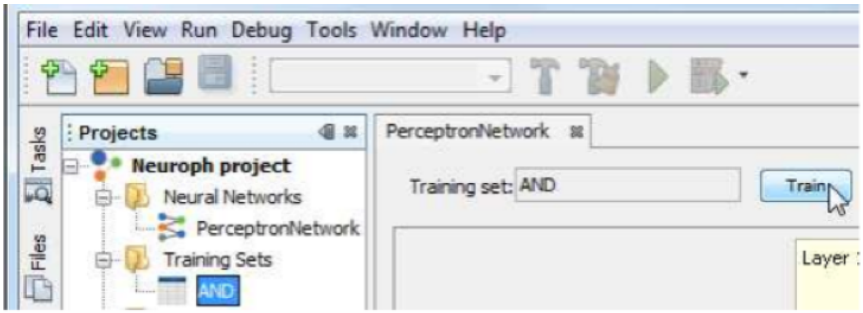


Fuente: propia

Paso 4: El entrenamiento de la red.

Para iniciar el procedimiento de formación de la red, en la ventana de la red seleccione conjunto de entrenamiento y haga clic en el botón de Train

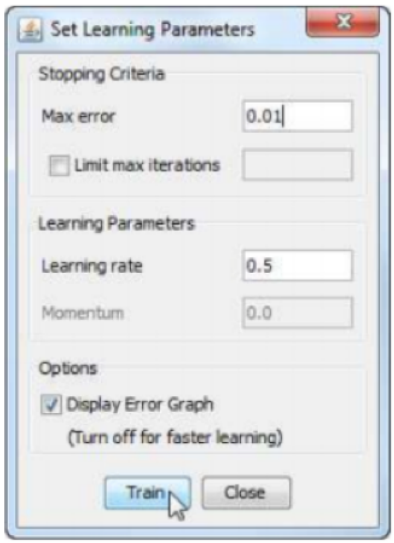
Ilustración 21. Iniciar el entrenamiento



Fuente: propia

En el cuadro de diálogo de los parámetros del Conjunto de Aprendizaje por defecto haga clic en el botón de Aprendizaje.

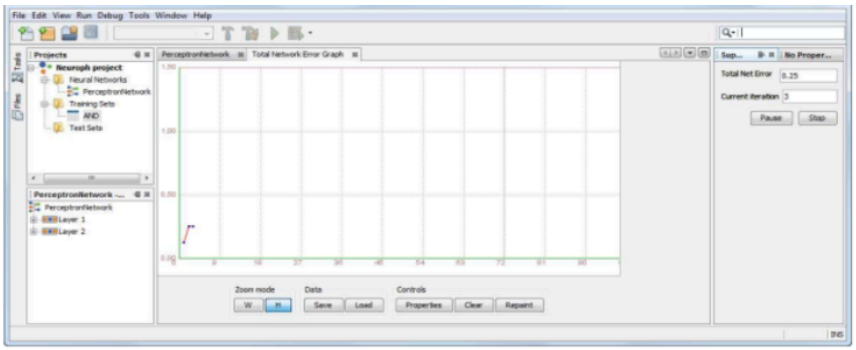
Ilustración 22. Parámetros para entrenamiento



Fuente: propia

Cuando el error neto total es cero, la formación es completa (que será muy rápido ya que este ejemplo es muy sencillo)

Ilustración 23. Indicadores de fin de entrenamiento.

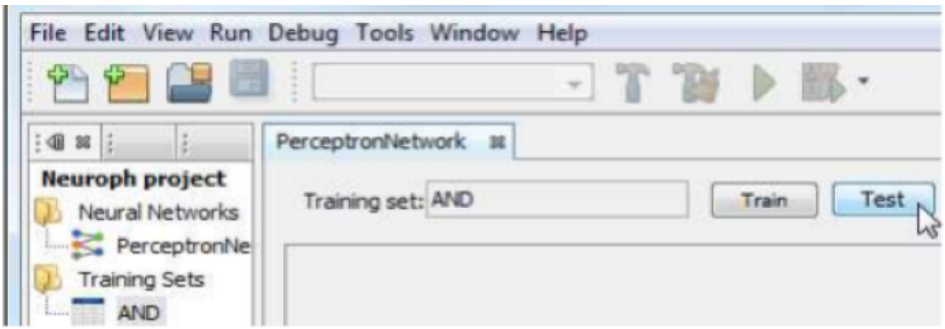


Fuente: propia

Paso 5: Probar la red.

Después que la formación se ha completado, puede probar conjunto de entrenamiento haciendo clic en el botón de prueba.

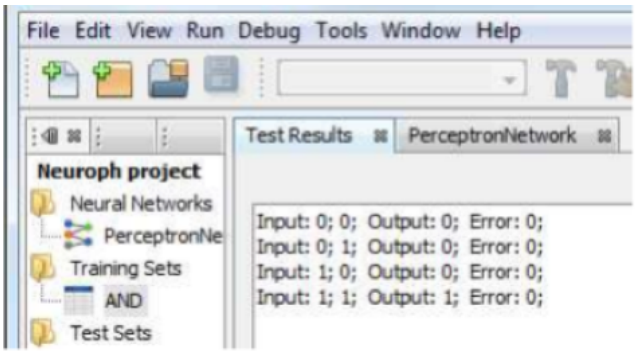
Ilustración 24. Prueba de la RNA



Fuente: propia

Esto abrirá resultados de prueba en una nueva pestaña.

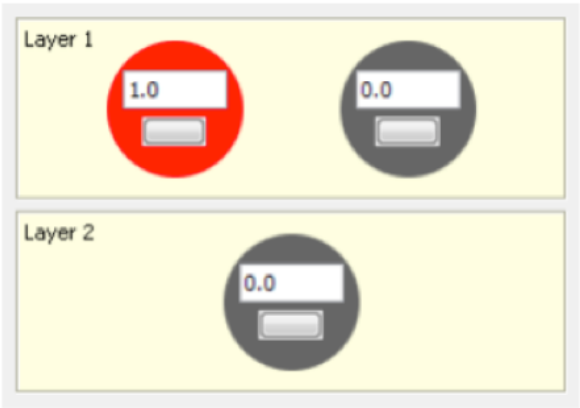
Ilustración 25.Resultados de prueba



Fuente: propia

La Red aprendió la función AND lógica. Como podemos ver la neurona de salida tiene un valor 0. Se prueba de la red para ver cómo se comporta para otros valores de entrada.

Ilustración 26. Verificación del funcionamiento correcto



Fuente: propia

Se puede seleccionar la vista de gráfico para ver diferentes vistas de la red.

Ilustración 27.Vista de gráfico

Fuente: (Sevarac & Koprivica, Neuroph, S/A)

JDOM

Hoy en día existe una gran cantidad de software que permiten acceder a un documento XML a través de una estructura de árbol, por ejemplo, JDOM el cual proviene de las siglas Java Document Object Model, sin embargo fue implementado exclusivamente para el lenguaje de programación java por lo que es relativamente fácil ya que permite a los usuarios que tienen conocimiento sobre el lenguaje de java desarrollar sobre esta API de una manera natural e intuitiva. Basándose en (O'REILLY, 2012) y (JDOM, 2013) nos indica que JDOM es una API de código abierto esto quiere decir que los usuarios de dicha API pueden dar sugerencias y para que la herramienta pueda seguir creciendo y a adaptarse a las nuevas necesidades.

Según (Harold) nos indica que con la API JDOM, para el procesamiento del XML es conocer los paquetes que contiene la herramienta org.jdom, org.jdom.adapters, org.jdom.input or.jdom.output.

En Servidor de aplicaciones, JDOM, con el que la aplicación XML puede ser desarrollado rápidamente, se utiliza para codificar y decodificar los datos transportados en formato XML. En primer lugar, encapsular el resultado consultado a un objeto Document (org.jdom.Document). (Yang, 2009)

El paquete org.jdom se representa el documento XML, Element representa el elemento del documento y la clase atribute representa los atributos de cada elemento, El paquete org.jdom.adapters contiene todas las clases adaptadoras, El paquete org..jdom.input contiene las clases builder los cuales permiten crear los documentos XML y por último el paquete org.jdom.output contiene las clases que permiten presentar lo que tenga la clase Document.

En conclusión, JDOM es una de las mejores herramientas para el manejo de archivos XML siempre y cuando se esté trabajando con el lenguaje de programación JAVA y está construido a partir de árboles que pueden ser recorridos en cualquier dirección.

Esta herramienta cuenta con su descarga libre para los usuarios, encontrando sus versiones en la página oficial del mismo http://www.jdom.org/downloads/ , la que actualmente está en auge en la 2.0.6.

Una de las utilidades del API JDOM es el poder leer, crear y manipular documentos XML de una forma mucho más sencilla y es muy interactiva con el usuario programador, a diferencia de las API DOM Y SAX que se crearon sin pensar en algo concreto, lo cual es muy bajo el porcentaje de su utilización.

Ejemplo con JDOM

Primero importar las librerías

- jdom.jar
- jaxen-core.jar
- jaxen-jdom.jar
- saxpath.jar

Parsear XML

Para que JDOM pueda parsear un XML se usa SAXBuilder. El cual permitirá generar un árbol de nodos

```
// Creamos el builder basado en SAX
SAXBuilder builder = new SAXBuilder();
// Construimos el arbol DOM a partir del fichero xml
Document documentJDOM = builder.build(new FileInputStream("/ruta_a_fichero/fichero.xml"));
```

Recorriendo el árbol de nodos

Una vez que se obtiene la variable Document se puede recorrer el árbol de nodos como sea necesario.

```
// Obtenemos la etiqueta raíz
Element raiz = doc.getRootElement();
// Recorremos los hijos de la etiqueta raíz
List<Element> hijosRaiz = raiz.getChildren();
for(Element hijo: hijosRaiz){
   // Obtenemos el nombre y su contenido de tipo texto
   String nombre = hijo.getName();
   String texto = hijo.getValue();

   System.out.println("\nEtiqueta: "+nombre+". Texto: "+texto);

   // Obtenemos el atributo id si lo hubiera
   String id = hijo.getAttributeValue("id");
   if(id!=null){
      System.out.println("\tId: "+id);
   }
}
```

XPath

JDOM («JDOM», s. f.) permite buscar las etiquetas Xpath de un XML mediante los siguientes métodos:

- selectSingleNode: Devuelve un solo valor
- selectNodes: Devuelve una lista de valores

```
// Buscamos una etiqueta mediante XPath
Element etiquetaHijaXP = (Element)XPath.selectSingleNode(doc, "/etiquetaPrincipal/etiquetaHija");
```

Para ver la aplicación de los componentes antes mencionado debe dirigirse a la sección de ejemplos los cuales utilizan estos métodos para manejar los datos generados por JOONE.

Tensorflow

Es una herramienta orientada a la construcción de modelos de redes neuronales, es de código abierto y permite el cálculo numérico basándose en gráficos de flujo de datos para un aprendizaje automático mediante un rango de tareas, desarrollada por un equipo de investigadores del "Google Brain". El sistema facilita la investigación sobre el aprendizaje automático siendo utilizado como parte de una estrategia en Big Data por Google.

TensorFlow es un sistema de segunda generación, posee una arquitectura es flexible permitiéndole asignar cargar de computo a uno o más CPUs o GPUs ya sean estos de una computadora de escritorio, dispositivo móvil con una API o servidor. Este software fue liberado por Google siendo de código abierto el 9 de noviembre del 2015 y se encunetra disponible en Linux de 64 bits, macOS, plataformas móviles como Android y iOS.

Esta herramienta aún se encuentra en desarrollo por el equipo de trabajo de Google es por ello que no se encuentra completa y se seguirá trabajando en ella hasta ampliarla y mejorarla.

Matlab

Es un software matemático orientado a la resolución de problemas técnicos y de ingeniería, donde su principal función es el diseño de productos y sistemas mediante cálculo matemático que son visualizados por los gráficos que implementar esta herramienta matemática. Los cálculos que se realiza con esta herramienta presentan una alta precisión y su traficación es avanzada permitiendo la visualización de datos en 2D y 3D.

Se puede argumentar que Matlab es un lenguaje de programación que permiten desarrollo aplicaciones que involucren operación de vectores, matrices entre otros cálculos matemáticos, este lenguaje permite incluir en el desarrollo de los proyectos programación orientada a objetos. Cuanta una amplia biblioteca donde están disponibles diversos algoritmos matemáticos y funcionalidades.

Matlab es usado en diversas ramas de la ingeniera ya que tiene varias funcionalidades como el aprendizaje de máquina, diseño y control de robots, procesamiento de imágenes, entre otros; características que lo convierten es un lenguaje ideal para la construcción de modelos con redes neuronales.

Este lenguaje es integrado al desarrollo de computación técnica ya que ofrece un entorno flexible que incluye una gran variedad de herramientas (toolbox) donde las que se pueden usar para redes neuronales son “image processing”, “neural network”, “daa acquisiton”, entre otros.

Referencia bibliográfica

Adam, D. (2015). OBJECT-ORIENTED PROGRAMMING AND. *Studies in Logic, Grammar and Rhetoric,* 293.

Garmendia, L., & Santos, M. (s.f.). Red neuronal difusa para un planificador de entornos GRID. 8. Obtenido de http://www2.caminos.upm.es/Departamentos/matematicas/Fdistancia/MAIC/CONGRESOS/JORNADAS%201/124%20Red%20Neuronal%20Difusa%20para%20un%20Planificador%20de%20Entornos%20G_205.pdf

Harold, E. R. (s.f.). *Java Processing XML with*. New York: Addison-Wesley. Obtenido de https://books.google.com.ec/books?id=Ys98M8A45G4C&printsec=frontcover&dq=JDOM&hl=es&sa=X&redir_esc=y#v=onepage&q=JDOM&f=false

JDOM, P. O. (2013). Obtenido de http://www.jdom.org/mission/index.html

Marrone, P. (2007). The Complete Guide. 142.

Neuroph. (S/A). *Neuroph: Java Neural Network Framework*. Obtenido de http://neuroph.sourceforge.net/

O'REILLY. (2012). *JAVA & XML*. Obtenido de http://docstore.mik.ua/orelly/xml/jxml/ch07_01.htm

Sevarac, Z. (Febrero de 2011). *Oracle*. Obtenido de http://www.oracle.com/technetwork/articles/java/nbneural-317387.html

Sevarac, Z., & Koprivica, M. (S/A). *Neuroph*. Obtenido de Java Neural NetworkFramework:http://neuroph.sourceforge.net/Getting%20Started%20with%20Neuroph%202.7.pdf

Yang. (2009). Power Electronics and Intelligent Transportation System (PEITS),.

Redes neuronales artificiales aplicadas al reconocimiento de patrones
Edición digital 2017-2018.
www.utmachala.edu.ec

Redes

Redes es la materialización del diálogo académico y propositivo entre investigadores de la UTMACH y de otras universidades iberoamericanas, que busca ofrecer respuestas glocalizadas a los requerimientos sociales y científicos. Los diversos textos de esta colección, tienen un espíritu crítico, constructivo y colaborativo. Ellos plasman alternativas novedosas para resignificar la pertinencia de nuestra investigación. Desde las ciencias experimentales hasta las artes y humanidades, Redes sintetiza policromías conceptuales que nos recuerdan, de forma empeñosa, la complejidad de los objetos construidos y la creatividad de sus autores para tratar temas de acalorada actualidad y de demanda creciente; por ello, cada interrogante y respuesta que se encierra en estas líneas, forman una trama que, sin lugar a dudas, inervará su sistema cognitivo, convirtiéndolo en un nodo de esta urdimbre de saberes.

UNIVERSIDAD TÉCNICA DE MACHALA
Editorial UTMACH
Km. 5 1/2 Vía Machala Pasaje
www.investigacion.utmachala.edu.ec / www.utmachala.edu.ec

ISBN: 978-9942-24-100-9