

Computación Blanda

Soft Computing

Autor: **Jose Daniel Velasquez Gordon**

IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia

Correo-e: j.velasquez@utp.edu.co

I.1 REDES NEURONALES

Resumen— Este documento presenta un resumen de las líneas clásicas de la Computación Blanda: redes neuronales, lógica difusa, sistemas expertos, algoritmos genéticos y machine learning. El objetivo del documento es brindar una panorámica general de las temáticas, mostrando su relación con las técnicas de inteligencia artificial. La diferencia entre el paradigma de Inteligencia Artificial y la computación blanda está centrada en el mecanismo de inferencia utilizado y su aplicación a la solución de problemas tomados de lo cotidiano, de las teorías de conocimiento y de su relación con ciencias afines.

Palabras clave— sistemas, redes, inteligencia artificial, software, computación, investigación, industria, genético, aprendizaje.

Abstract— This document presents a summary of the classic lines of Soft Computing: neural networks, fuzzy logic, expert systems, genetic algorithms and machine learning. The objective of the document is to provide a general overview of the topics, showing their relationship with artificial intelligence techniques. The difference between the Artificial Intelligence paradigm and soft computing is centered on the inference mechanism used and its application to the solution of problems taken from everyday life, from knowledge theories and their relationship with related sciences.

Key Word— systems, networks, artificial intelligence, software, computing, research, industry, genetic, learning.

I. INTRODUCCIÓN

La temática de la Computación Blanda se encuentra enmarcada en el paradigma de la Inteligencia Artificial. La diferencia con dicho paradigma radica en que la Computación Blanda está centrada en la aplicación pragmática de las teorías de la Inteligencia Artificial a la solución de problemas complejos en diversos campos del conocimiento.

Las líneas derivadas de la Computación Blanda se configuran en las siguientes tendencias: a) Redes Neuronales Artificiales, b) Lógica Difusa, c) Sistemas Expertos, d) Algoritmos Genéticos, e) Deep Learning (Machine Learning).

En los siguientes apartados se presenta un resumen de dichas tendencias.

Las redes neuronales se podrían definir como un modelo de las redes neuronales biológicas, aquellas que hacen posible el pensamiento y raciocinio en las personas. A pesar de que se les describe como un modelo biológico, en la parte técnica difiere mucho de este aspecto, ya que las redes neuronales están cimentadas en la matemática y la estadística, aunque el resultado final se espera que sea parecido a las redes neuronales biológicas, la forma en que funcionan difieren mucho.

Empezando por lo más básico de las redes neuronales, se puede hablar del perceptrón, este se asemeja a una sola neurona artificial. El Perceptrón es un elemento que consta de dos entradas o más, una combinación de estas y una salida. A las entradas se les asigna un peso el cual va a ser muy importante ya que en la combinación que es una suma, si la multiplicación de las entradas con sus pesos, más la suma de todo esto da mayor que un número determinado la salida es uno, de otro modo ósea cuando es menor da cero.

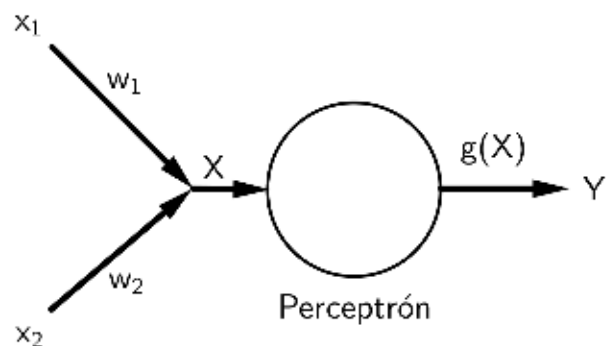


Ilustración 1. Sacado de GeoGebra. Donde los pesos son w_1 y w_2 , las entradas x_1 y x_2 , y la salida Y .

La mejor forma de saber cómo funciona un Perceptrón es con un ejemplo (este ejemplo se sacó de [1]). Supongamos que son alumnos de una clase en la que el profesor no ha dicho exactamente cómo va a poner las notas. Para empezar, supongamos que sólo se han hecho dos exámenes y se tiene la nota de cada uno de ellos y la final. Las entradas del Perceptrón serían las dos notas de los exámenes, Si la salida es uno (esto es, la suma de las notas por su peso correspondiente es mayor que cinco), es un aprobado. Si es cero, suspenso. Los pesos son lo que

tenemos que encontrar con el entrenamiento. En este caso, nuestro entrenamiento consistirá en empezar con dos pesos aleatorios (por ejemplo, 0.5 y 0.5, el mismo peso a cada examen) y ver qué resultado da la red neuronal para cada alumno. Si falla en algún caso, iremos ajustando los pesos poco a poco hasta que esté todo bien ajustado. Por ejemplo, si un alumno con muy buena nota en el segundo examen ha suspendido el curso, bajaremos el peso del segundo examen porque claramente no influye demasiado. Poco a poco acabaremos encontrando los pesos que se ajusten a las notas que puso el profesor. La idea del ajuste o retroalimentación es ir adaptando la red a la información "oculta" que tienen los datos que le pasamos para que aprenda.

Esto es un ejemplo muy ejemplo muy simple de red neuronal, pero si nos vamos a la definición original nos damos cuenta de que nuestro cerebro funciona con una red de neuronas, esto en las redes neuronales artificiales se les llama Capas y es un paso más allá para resolver problemas más complejos. Como hicimos anteriormente la mejor forma de ver cómo se comporta una red neuronal de capas es por medio de un ejemplo (este ejemplo se sacó de [1]).

Hay dos alumnos que tienen la misma nota en los exámenes, dos dieces, pero uno tiene un 7 en el trabajo y otro un 4. El del 7 ha aprobado el curso, pero el del 4 no. Hay un alumno que tiene un 10 en el trabajo y 4.99 en los dos exámenes y que está suspenso. Podemos intentar entrenar una red neuronal como la de antes todo lo que queramos en esta situación que no va a funcionar bien. Es posible que funcione en muchos casos, pero no va a ser perfecta. Y es que parece que la nota del trabajo no influye salvo que lo suspenda, en cuyo caso este suspenso directamente. Es un filtro, un uno o un cero que tenemos que sacar en la red neuronal antes de poder dar el resultado de aprobado o suspendido en el curso.

Efectivamente: necesitamos más capas. Necesitamos un perceptrón intermedio que nos diga si el trabajo está aprobado o no, y contar eso en el perceptrón de salida.

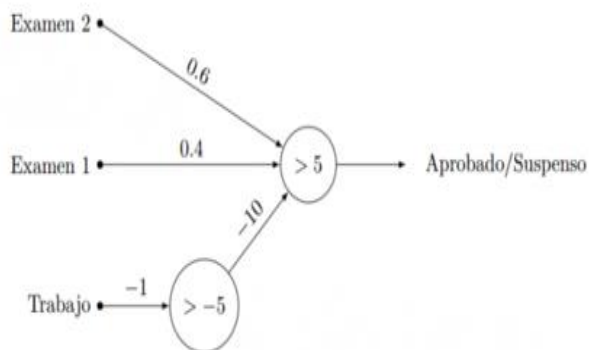


Ilustración 2. Sacada de xataca.

El primer perceptrón mira si la nota del trabajo multiplicada por menos uno es mayor que menos cinco. Si lo es, entonces su salida es uno. Al multiplicarla por menos diez en la entrada del segundo perceptrón, forzará siempre un suspenso. Si el trabajo está aprobado, la salida del primer perceptrón será 0 y no afectará a la

media de los exámenes. Lo que hemos logrado con esta capa demás es añadir más información al problema, en ese caso que pasaba si el trabajo estaba aprobado o no. La capa que se utilizó para este ejemplo es básica, pero en redes neuronales robustas, por lo general las capas están compuestas por varios nodos de perceptrones y estas están conectadas a más capas.

Con el tiempo las redes neuronales han avanzado más, por ejemplo, en algunas se dejan de usar perceptrones para utilizar un tipo de neurona llamada sigmoide, esta neurona que se basa en la gráfica sigmoide, se podría decir que es más suave que el Perceptrón ya que los pesos no afectan de una forma tan abrupta.

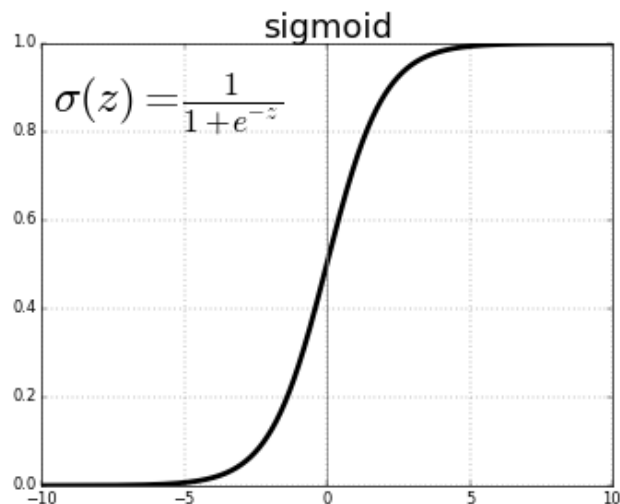


Ilustración 3. Sacada de [2].

Donde $z = b + \sum_i w_i x_i$

Por otro lado, tenemos las redes neuronales profundas, estas redes tienen más capas intermedias, por lo cual se usan para tratar datos más complejos y mejorar el resultado de aprendizaje. Por último tenemos las redes convolucionales, que están funcionando muy bien en reconocimiento de voz y procesamiento de imágenes.

I.2 LÓGICA DIFUSA

Una de las formas de definir la lógica difusa de una forma no tan formal es decir que es la lógica que tiene en cuenta el medio entre dos extremos. Es la lógica que utiliza expresiones que no son ni totalmente ciertas ni completamente falsas, es decir, es la lógica aplicada a conceptos cualquiera que pueden tomar un valor cualquiera dentro de un conjunto de valores que oscilan entre dos extremos, la verdad absoluta y la falsedad total [3].

Este tipo de lógica se creó con la idea en mente de que el mundo real es mucho más complejo que dos simples valores como 1 y 0, por lo que se necesitaba un sistema formal más preciso de razonamiento.

Para entender un poco más de la lógica difusa se puede hablar de los conjuntos, los cuales también están presentes en la lógica clásica.

El primero en hablar acerca de conjuntos difusos fue Lofti A. Zadeh en un artículo que se llama Fuzzy Sets, en este propuso el problema de los hombres altos en el cual no había una visión clara de quienes eran altos y quienes no como se muestra en la Ilustración 4.

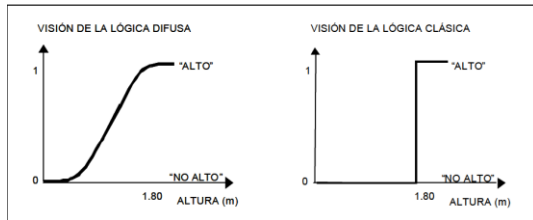


Ilustración 4.

Para esto propuso y definió los conjuntos difusos de la siguiente forma.

“Sea X un conjunto clásico. Un conjunto difuso, A , en X viene caracterizado por la función de pertenencia $f_A(x)$, que asocia a cada punto $x \in X$ un número real del intervalo $[0,1]$, donde los valores de $f_A(x)$ representan el "grado de pertenencia" de x en A , de forma que, cuanto más cerca esté el valor de $f_A(x)$ a 1, mayor es el grado de pertenencia de x a A .”[4]

Esta definición no deja de lado los conjuntos clásicos ya que los conjuntos difusos se pueden considerar como una extensión de los clásicos ya que están dentro de estos donde si entendemos la definición nos damos cuenta de que por medio de a función de pertenencia también se puede describir un conjunto clásico. además de eso muchas de las reglas que se aplicaban antes en los conjuntos clásicos también se pueden aplicar a los difusos.

En cuanto a la función $f_A(x)$, se tiene que elegir una función que se adapte mas al problema, con la caracteriztica que debe tomar valores entre 0 y 1, con continuidad. Para esto las mas utilizadas son el triangular, trapezoidal, gaussiana, sigmoideal, gamma, pi campana, etc.

Existen también lo que es operaciones con conjuntos difusos, existen operaciones básicas que cumplen con la asociatividad, conmutatividad y distributivita como las leyes de Morgan.

El conjunto complementario $-A$ de un conjunto difuso viene dado por su función característica:

$$f_{-A}(x) = 1 - f_A(x)$$

La Unión de dos conjuntos A y B es un conjunto difuso $A \cup B$ en U cuya función de pertenencia es la siguiente:

$$f_{A \cup B}(x) = \max [f_A(x), f_B(x)]$$

La intersección de dos conjuntos difusos A y B es un conjunto difuso $A \cap B$ en U con función característica:

$$f_{A \cap B}(x) = \min [f_A(x), f_B(x)]$$

Hay otras reglas que no se cumplen en los conjuntos difusos como el principio de contradicción y el de exclusión.

también existe Inferencia e implicación Difusa, para esto las reglas difusas son un conjunto de proposiciones IF-THEN, que es el modelo del problema que se quiere resolver. Las reglas son de la forma:

“sí u es A entonces v es B ”

Donde A y B son conjuntos difusos definidos en los rangos u y V . la función característica de estas proposiciones son $f_{A \rightarrow B}(x)$.

La lógica difusa actualmente se usa en muchas áreas, este solo fue un pequeño resumen de toda la teoría acerca de lógica difusa.

I.3 SISTEMAS EXPERTOS

Los sistemas expertos son lo más parecido a un experto en materia, estos trabajan sobre un área muy específica en la cual se desempeñan muy bien. Se pueden definir como un sistema de información basado en el conocimiento que usa su conocimiento de un área de aplicación compleja y específica a fin de actuar como un consultor experto para los usuarios finales [5]. Este es un derivado de la rama principal inteligencia artificial.

Los sistemas expertos simulan gran parte del comportamiento humano como aprendizaje, memorización, razonamiento, comunicación. también se caracteriza por 4 puntos muy importantes.

- Habilidad para adquirir conocimiento: Esto como cualquier experto debe tener unos conocimientos previos que se le dan al sistemas o mejor llamado base de conocimiento.
- Fiabilidad, para poder confiar en sus resultados o apreciaciones: Uno de los aspectos más importantes de los sistemas expertos es poder justificarse a ellos mismos a sus decisiones, explicar porque tomo este camino y este no.
- Solidez en el dominio de su conocimiento: Con buenas bases de conocimiento se mejoran los resultados.
- Capacidad para resolver problemas.

Las partes que componen a un sistema experto son las siguientes.

- Base de datos: almacenamiento de información en forma estructurada para su posterior manejo de forma dinámica.
- Base de conocimientos: Hechos y reglas, por lo general de expertos en el tema.
- Motor de inferencia: Es la forma como el sistema experto resuelve un problema. O componente del sistema que aplica las reglas, seleccionando aquellas cuyas premisas coinciden con hechos -en lo que se conoce como «encadenamiento hacia adelante»- o seleccionando aquellas cuyas conclusiones coinciden con el fin buscado -en lo que se conoce como «encadenamiento hacia atrás»- para poder así verificar las premisas [6].
- Interfaz de usuario o de comunicación: Es la interfaz por la cual el usuario se puede comunicar y por el cual el sistema experto puede dar a conocer los resultados como su razonamiento. Todo esto mediante un lenguaje natural.

Existen tres tipos de sistemas expertos.

Basados en reglas: Este tipo de sistema es el más simple y

rígido, sirve para resolver problemas preferiblemente determinísticos, su funcionamiento depende mucho de las reglas de su base de conocimientos.

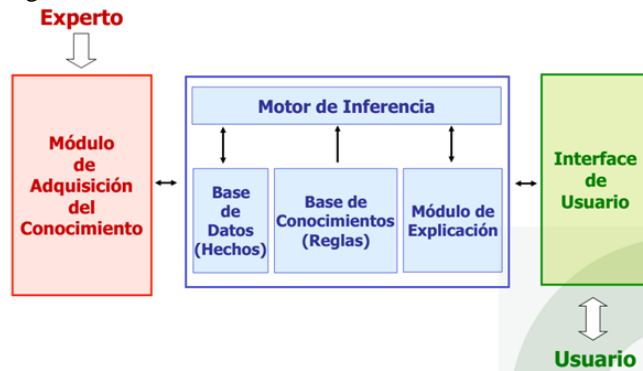


Ilustración 5. Arquitectura de un Sistema Experto Basado en Reglas.

Basados en casos: La solución de un problema similar planteado con anterioridad se adapta al nuevo problema. Este tipo de sistema experto es muy poderoso, ya que trabaja como la mente humana, con la experiencia de vivencias previas, además las reglas se van adaptando a estas.

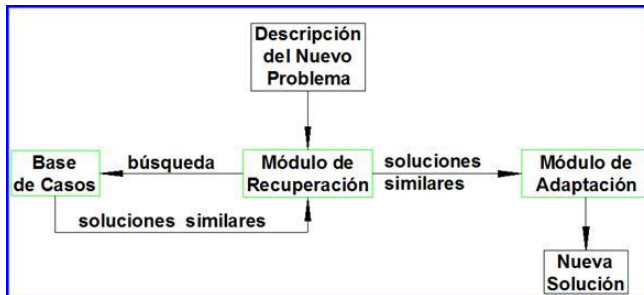


Ilustración 6. Sistema experto basado en casos.

Basado en redes: Consiste en un gráfico que representa un conjunto de variables conocidas y la relación entre ellas a fin de inferir la probabilidad de las variables no conocidas.

Los sistemas expertos tienen sus ventajas y desventaja, por ejemplo, los sistemas expertos no son buenos trabajando en equipo o investigando algo nuevo, son costosos y la falta de contexto en problemas. Por otro lado, son confiables, pueden trabajar una gran cantidad de horas, etc.

I.4 ALGORITMOS GENÉTICOS

Charles Darwin fue uno de los personajes más importantes de la historia, gracias a él tuvimos una visión más detallada de como ocurrió la evolución. De esta teoría salió una idea muy importante que “el azar en la variación, junto con la ley de selección, es una técnica de resolución de problemas de inmenso poder y de aplicación casi ilimitada.”[7]. Con esto en mente podemos empezar a hablar de una de las herramientas más poderosas de la inteligencia artificial, los algoritmos genéticos.

Los Algoritmos genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los

organismos vivos [8]. Estos algoritmos para resolver un problema, como entrada se le debe pasar un conjunto de soluciones potenciales a ese problema, codificadas y una función que nos ofrece una métrica, esta función se le llama función de aptitud o fitness en algunos casos, así nos da una medida cuantitativa de las soluciones.

La función de aptitud evalúa cada solución, al principio la mayoría tiene una eficiencia mínima y se van descartando por lo que por medio del azar quedan solo unas soluciones. De ahí el algoritmo lo que hace es reproducir estas soluciones haciendo copias, la diferencia es que a estas copias se les introducen cambios aleatorios (mutación), luego estas son el nuevo conjunto de soluciones candidatas, se vuelve a utilizar la función de aptitud y se eliminan aquellas que no han mejorado o por el contrario han empeorado. Lo que pasa en el algoritmo es una iteración de ese proceso cuantas veces sea necesario para llegar a la solución necesario o especificada.

```

BEGIN /* Algoritmo Genetico Simple */
  Generar una poblacion inicial.
  Computar la funcion de evaluacion de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generacion */
      FOR Tamaño poblacion/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generacion,
          para el cruce (probabilidad de seleccion proporcional
          a la funcion de evaluacion del individuo).
          Cruzar con cierta probabilidad los dos
          individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Computar la funcion de evaluacion de los dos
          descendientes mutados.
          Insertar los dos descendientes mutados en la nueva generacion.
        END
      END
      IF la poblacion ha convergido THEN
        Terminado := TRUE
      END
    END
  END

```

Ilustración 7. Algoritmo genético simple [8].

Antes de que un algoritmo genético se ejecute se tienen que codificar las posibles soluciones para que el algoritmo pueda trabajar con estas, para ello muchas veces se utiliza el sistema binario, que codifica las soluciones como cadenas binarias. Hay otra forma también de codificar más compleja pero a la vez mucho más precisa que con binarios, esta se da representando las soluciones en enteros o decimales, la ventaja es que tiene mayor precisión y puede con problemas de mayor complejidad. Todo esto se hace para que a la hora de inyectar el factor aleatorio sea más fácil. Por último tenemos la forma de John Koza, donde representa los programas como estructuras de datos ramificadas o árboles. Para hacer el cambio aleatorio se altera el valor de un cierto nodo, o sustituyendo una parte del árbol (subárbol) por otra.

El método de selección de la siguiente generación es muy importantes, ya que con malos criterios de selección nunca se llegaría a la solución deseada, para esto existe un gran conjunto de consideraciones a la hora de elegir la próxima generación.

- Selección elitista: La selección de los miembros más aptos de cada generación. Por lo general se utiliza con otros criterios.

- Selección proporcional a la aptitud: Los individuos más aptos tienen más probabilidades de ser seleccionados, pero no la certeza.[7]
- Selección por rueda de ruleta: probabilidad de que un individuo sea seleccionado es proporcional a la diferencia entre su aptitud y la de sus competidores.[7]
- Selección escalada: cuando se incrementa la aptitud media de una población, se fuerza la presión selectiva, y se vuelve más discriminadora la función de aptitud. Por lo general este criterio se utiliza en las últimas etapas del algoritmo cuando todas las soluciones tienen una aptitud relativamente alta.
- Selección por torneo: Se hacen subgrupos, compiten entre ellos y se elige el mejor.
- Selección por rango: a cada individuo de la población se le asigna un rango numérico basado en su aptitud, y la selección se basa en este ranking, en lugar de las diferencias absolutas en aptitud. La ventaja de este método es que puede evitar que individuos muy aptos ganen dominancia al principio a expensas de los menos aptos, lo que reduciría la diversidad genética de la población y podría obstaculizar la búsqueda de una solución aceptable [7].
- Selección jerárquica: En cada etapa los individuos atraviesan rondas de selección, siendo las menos rigurosas las del principio, mientras que al final del algoritmo son más rigurosas. Esto se hace para acotar tiempo.

Otro aspecto muy importante del algoritmo es el método de cambio, en el cual vamos a cambiar la población y generar nuevos descendientes.

Existen Dos tipos de métodos de cambio. El primero de estos es mutación, es el método más simple y se cambian pequeñas porciones de la codificación. Por otro lado, tenemos el cruzamiento, consiste en dos individuos que se seleccionan para que intercambien varios segmentos de su código, cuya descendencia sería una combinación de los padres.

Existen algoritmos más complejos y con más variaciones, este solo fue un vistazo rápido a las bases de los algoritmos genéticos. Estos se utilizan en un gran ámbito en la sociedad por su utilidad.

I.1 DEEP LEARNING

Deep learning esta muy asociado al primer tema redes neuronales y al uso que se les dan a estas. Deep learning es una tecnica de machine learning basado en redes neuronales. Se puede definir como un algoritmo automático estructurado o jerárquico que emula el aprendizaje humano.

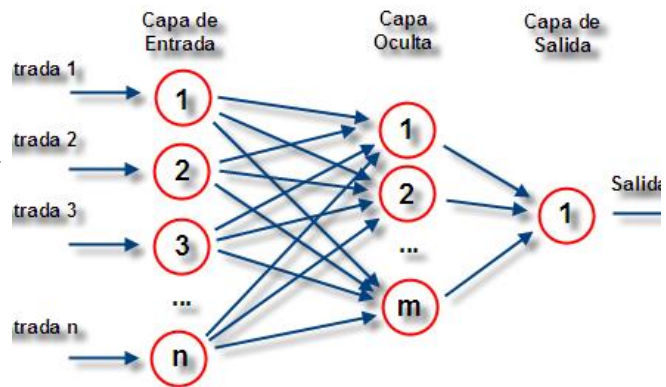


Ilustración 9. Red neuronal.

Entendiendo ya como funcionan las redes neuronales de forma básica, no se centrará mucho en cómo funciona un sistema de estos, ya que deep learning son prácticamente redes neuronales, la diferencia radica en el uso que se le dan.

Uno de los grandes avances del deep learning es que además de que el sistema se puede entrenar como en machine learning, este puede entrenarse así mismo, y así encontrar coherencia en datos aleatorios. En pocas palabras el aprendizaje autónomo.

Hay varios ambitos en los cuales el deep learning se está haciendo espacio, el más reconocido de ellos es el de reconocimiento de imágenes, esto lo hace mediante diferentes capas intermedias de redes neuronales, mientras más capas mejor el resultado, además cabe destacar que este tipo de algoritmos funcionan con aprendizaje supervisado y no supervisado.

Otros de los grandes escenarios que está dando de qué hablar es el de conducción autónoma, como los coches de Tesla, con el pasar del tiempo la conducción autónoma se ha refinado de una manera impresionante, volviéndolo ahora mucho más confiable y viable.

Hay otros campos como lenguaje natural hablado y escrito, reconocimiento de voz, interpretación semántica, reconocimiento facial, etc.

REFERENCIAS

Referencias en la Web:

[1] <https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo>

[2] https://ml4a.github.io/ml4a/es/neural_networks/#:~:text=Una%20neurona%20que%20utiliza%20la, trav%C3%A9s%20de%20la%20funci%C3%B3n%20sigmoide.

[3] <https://www.tdx.cat/bitstream/handle/10803/6887/04Rpp04de11.pdf>

[4] <https://www.cs.us.es/~fsancho/?e=97>

[5]

https://www.ecured.cu/Sistemas_expertos

[6]

<file:///C:/Users/jose/Downloads/Dialnet-IntroduccionALosSistemasExpertos-257148.pdf>

[7]

<https://www.cs.us.es/~fsancho/?e=65>

[8]

<http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/emageneticos.pdf>

<file:///C:/Users/jose/Downloads/Dialnet-IntroduccionALosSistemasExpertos-257148.pdf>