

Computación Blanda

Soft Computing

Autor: **Steven Medina Gonzalez**

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

Correo-e: steven.medina@utp.edu.co

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.

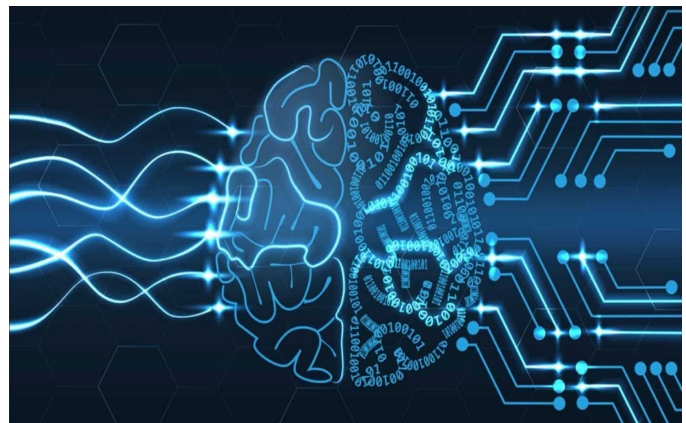
Las líneas derivadas de la Computación Blanda, se configuran en las siguientes tendencias: a) Redes Neuronales Artificiales, b) Lógica Difusa, c) Sistemas Expertos, d) Algoritmos Genéticos, e) Deep Learning (Machine Learning).

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

1.1 REDES NEURONALES

Las redes neuronales artificiales (también conocidas como sistemas conexionistas) son un modelo computacional vagamente inspirado en el comportamiento observado en su homólogo biológico. Consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida.

Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Del mismo modo, a la salida de la neurona, puede existir una función limitadora o umbral, que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona. Esta función se conoce como función de activación.

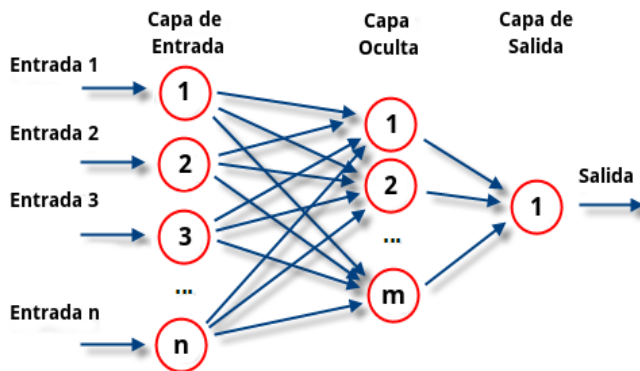


Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, y sobresalen en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional. Para realizar este aprendizaje automático, normalmente, se intenta minimizar una función de pérdida que evalúa la red en su total. Los valores de los pesos de las neuronas se van actualizando

buscando reducir el valor de la función de pérdida. Este proceso se realiza mediante la propagación hacia atrás.

Como se ha mencionado el funcionamiento de las redes se asemeja al del cerebro humano. Las redes reciben una serie de valores de entrada y cada una de estas entradas llega a un nodo llamado neurona. Las neuronas de la red están a su vez agrupadas en capas que forman la red neuronal. Cada una de las neuronas de la red posee a su vez un peso, un valor numérico, con el que modifica la entrada recibida. Los nuevos valores obtenidos salen de las neuronas y continúan su camino por la red. Este funcionamiento puede observarse de forma esquemática en la siguiente imagen.

El objetivo de la red neuronal es resolver los problemas de la misma manera que el cerebro humano, aunque las redes neuronales son más abstractas. Las redes neuronales actuales suelen contener desde unos miles a unos pocos millones de unidades neuronales, además de que otro objetivo de estas redes es aprender modificándose automáticamente a si mismo de forma que puede llegar a realizar tareas complejas que no podrían ser realizadas mediante la clásica programación basada en reglas. De esta forma se pueden automatizar funciones que en un principio solo podrían ser realizadas por personas.



Las redes neuronales se han utilizado para resolver una amplia variedad de tareas, como la visión por computador y el reconocimiento de voz, que son difíciles de resolver usando la ordinaria programación basado en reglas. Históricamente, el uso de modelos de redes neuronales marcó un cambio de dirección a finales de los años ochenta de alto nivel, que se caracteriza por sistemas expertos con conocimiento incorporado en si-entonces las reglas, a bajo nivel de aprendizaje automático, caracterizado por el conocimiento incorporado en los parámetros de un modelo cognitivo con algún sistema dinámico.

Las redes neuronales no son una idea nueva. Datan de los años 40 y 50, cuando se empezaron a publicar los primeros conceptos. Sin embargo, nunca tuvieron un gran éxito, más que nada porque se necesita una cantidad importante de recursos de un ordenador para entrenar y ejecutar una red neuronal con buenos resultados.

Las redes neuronales parece que incluso podrían acabar dominando uno de los juegos que se les resiste a los ordenadores: el juego de Go. En la Universidad de Edimburgo, unos investigadores han logrado usar redes convolucionales para detectar patrones en los tableros y tratar de sacar el mejor movimiento con una efectividad considerable: 90% de juegos ganados contra GNU Go y 10% contra Fuego, dos de los programas que mejor juegan a Go. Aunque pueda parecer poco, hay que tener en cuenta que ambos exploran un buen número de movimientos posibles para ver cuál da más ventaja. La red neuronal sólo mira al estado actual del tablero y emite un veredicto en muchísimo menos tiempo.

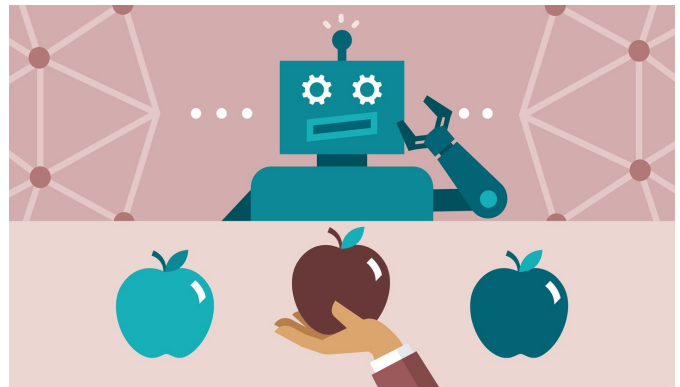
1.2 LÓGICA DIFUSA

La forma en que la gente piensa es, inherentemente, difusa. La forma en que percibimos el mundo está cambiando continuamente y no siempre se puede definir en términos de sentencias verdaderas o falsas.

Lógica difusa o lógica heurística se basa en lo relativo de lo observado como posición diferencial. Este tipo de lógica toma dos valores aleatorios, pero contextualizados y referidos entre sí.

Este tipo de lógica Se basa en dar solución a los problemas de exactitud ya que posee técnicas para dar problemas de incertidumbre con respecto a la precisión de datos que analicemos. Utilizando la teoría de los conjuntos el cual argumenta que un miembro de un conjunto puede pertenecer a más de un conjunto con cierto grado de pertenencia.

El objetivo de la lógica difusa sería crear un valor de la proposición de entrada en un valor de salida ya sea verdadero o falso, la diferencia sería que, con una mayor precisión, un ejemplo difuso sería saber si un vaso que esté lleno a la mitad esta medio lleno o medio vacío.



Consideremos como ejemplo el conjunto de vasos del mundo, que pueden estar vacíos o llenos de agua. Ahora tomemos un vaso vacío y comencemos a echar agua poco a poco, ¿en qué momento decidimos que el vaso pasa de estar vacío a estar

llo? Evidentemente, hay dos situaciones extremas que reconocemos sin ninguna duda, la primera cuando el vaso está completamente vacío, sin una sola gota de agua en su interior, y la segunda cuando está completamente lleno, cuando no cabe ni una sola gota más en él, pero una gota antes de estar completamente lleno, ¿diríamos que es falso que el vaso está lleno?, observa que para afirmar su condición, en la frase anterior no solo he usado el término lleno, sino que he añadido un modificador diciendo completamente lleno. Si a un vaso lleno de agua le quito una gota de agua... ¿dejo de considerarlo lleno y automáticamente pertenece para mí a otra clasificación?

Las definiciones de vaso completamente vacío y vaso completamente lleno son demasiado estrictas como para que resulten interesantes en un razonamiento en el que se consideran operaciones de llenado y vaciado de vasos, y entre los términos de lleno y vacío hay un área que no está claramente definida de pertenencia a ninguno de esos extremos. En el lenguaje natural que usamos en el mundo real hemos cubierto esta imprecisión por medio de una jerarquía de términos intermedios junto con modificadores que permiten cubrir un espectro más grande de áreas usando un número limitado de ellos, y podemos hablar de lleno, medio lleno, completamente lleno, casi lleno, etc. Matemáticamente, los conceptos de sí/no, verdadero/falso están representados por medio del concepto clásico de conjunto, pero necesitamos extenderlo para poder representar este tipo de información más difusa.

Un conjunto difuso permite a sus elementos tener un grado de pertenencia. Si el valor 1 se asigna a los elementos que están completamente en el conjunto, y 0 a los que están completamente fuera, entonces los objetos que están parcialmente en el conjunto tendrán un valor de pertenencia estrictamente entre 0 y 1. Por tanto, si un vaso completamente lleno tiene un grado de pertenencia a los vasos llenos de valor 1, y un vaso completamente vacío un grado de pertenencia a los vasos llenos de valor 0, entonces al añadir una gota a este último, su grado de pertenencia a los vasos llenos sería ligeramente superior a 0.



Se adapta mejor al mundo real en el que vivimos, e incluso puede comprender y funcionar con nuestras expresiones, del

tipo "hace mucho calor", "no es muy alto", "el ritmo del corazón está un poco acelerado", etc.

La clave de esta adaptación al lenguaje, se basa en comprender los cuantificadores de nuestro lenguaje, en los ejemplos de arriba "mucho", "muy" y "un poco". En la teoría de conjuntos difusos se definen también las operaciones de unión, intersección, diferencia, negación o complemento, y otras operaciones sobre conjuntos en los que se basa esta lógica.

Se basa en reglas heurísticas de la forma SI (antecedente) ENTONCES (consecuente), donde el antecedente y el consecuente son también conjuntos difusos, ya sea puros o resultado de operar con ellos. Sirvan como ejemplos de regla heurística para esta lógica:

- SI hace muchísimo calor ENTONCES disminuyo drásticamente la temperatura.
- SI voy a llegar un poco tarde ENTONCES aumento levemente la velocidad.

Los métodos de inferencia para esta base de reglas deben ser simples, veloces y eficaces. Los resultados de dichos métodos son un área final, fruto de un conjunto de áreas solapadas entre sí, donde cada área es resultado de una regla de inferencia. Para escoger una salida concreta a partir de tanta premisa difusa, el método más usado es el del centroide, en el que la salida final será el centro de gravedad del área total resultante.

1.3 SISTEMAS EXPERTOS

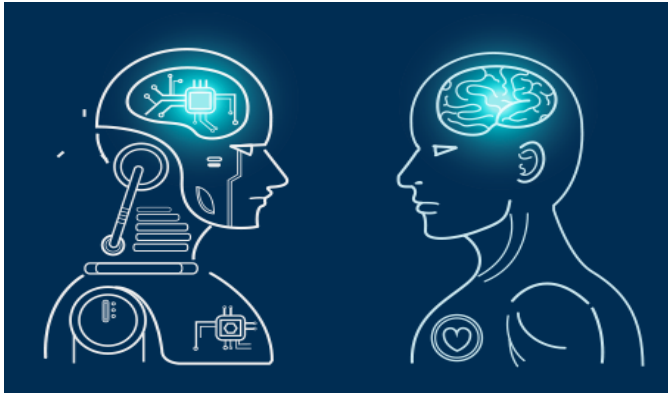
Los sistemas expertos es un sistema de información basado en el conocimiento que usa su conocimiento de un área de aplicación compleja y específica a fin de actuar como un consultor experto para los usuarios finales. Los sistemas expertos proporcionan respuestas sobre un área problemática muy específica al hacer inferencias semejantes a las humanas sobre los conocimientos obtenidos en una base de conocimientos especializados.

Para que un sistema actúe como un verdadero experto, es deseable que reúna, en lo posible, lo más importante de las características de un experto humano, esto es:

- Habilidad para adquirir conocimiento.
- Fiabilidad, para poder confiar en sus resultados o apreciaciones.
- Solidez en el dominio de su conocimiento.
- Capacidad para resolver problemas.

Dada la complejidad de los problemas que usualmente tiene que resolver un sistema experto, puede existir cierta duda en el usuario sobre la validez de respuesta obtenida. Por este motivo, es una condición indispensable que un sistema

experto sea capaz de explicar su proceso de razonamiento o dar razón del por qué solicita tal o cual información o dato. Estas características le permiten almacenar datos y conocimiento, sacar conclusiones lógicas, tomar decisiones, aprender de la experiencia y los datos existentes, comunicarse con expertos humanos, explicar el porqué de las decisiones tomadas y realizar acciones como consecuencia de todo lo anterior. Técnicamente un sistema experto, contiene una base de conocimientos que incluye la experiencia acumulada de expertos humanos y un conjunto de reglas para aplicar esta base de conocimientos en una situación particular que se le indica al programa. Cada vez el sistema se mejora con adiciones a la base de conocimientos o al conjunto de reglas.



Con su ayuda, personas con poca experiencia pueden resolver problemas que requieren un "conocimiento formal especializado". Se pueden obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos. Estos sistemas razonan, pero en base a un conocimiento adquirido y no tienen sitio para la subjetividad. Se ha comprobado que tienen al menos, la misma competencia que un especialista humano.

- Su uso es especialmente recomendado en las siguientes situaciones:
- Cuando los expertos humanos en una determinada materia son escasos.
- En situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas.

Cuando es muy elevado el volumen de datos que ha de considerarse para obtener una conclusión.

Por tanto, un sistema experto toma hechos y heurísticas para resolver problemas complejos de toma de decisiones. En ese orden, la calidad mejorada de las decisiones, la reducción de costos, la consistencia, la confiabilidad, la velocidad son los beneficios clave de un sistema experto.

La IA simula estos procesos y cuando hablamos de sistemas expertos nos referimos a la resolución de los problemas más complejos en un dominio específico, en otras palabras, el software de IA tiene el suficiente conocimiento almacenado como para resolver problemas complejos que solo un experto humano podría resolver.

Sin embargo, se debe tomar en cuenta que un sistema experto no puede dar soluciones creativas y su mantenimiento puede ser costoso. Fuera de eso, sus aplicaciones son muy amplias y son muy útiles para asegurar información rápida y precisa.

1.4 ALGORITMOS GENÉTICOS

Los Algoritmos Genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural y la supervivencia de los más fuertes, es decir la misma que subyace a la teoría de la evolución que formuló Charles Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real, así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven o sea mejores soluciones, y cuáles los menos aptos, que son descartados ósea las soluciones menos adecuadas.



Estos algoritmos hacen evolucionar una población de individuos someténdola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

Los algoritmos genéticos se enmarcan dentro de los algoritmos evolutivos, que incluyen también las estrategias evolutivas, la programación evolutiva y la programación genética.

Brevemente, un algoritmo genético (AG) es una técnica de resolución de problemas que imita a la evolución biológica como estrategia para resolver problemas, englobándose dentro de lo que antes hemos denominado técnicas basadas en poblaciones. Dado un problema específico a resolver, la

entrada del AG es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y una métrica llamada función de aptitud, o fitness, que permite evaluar cuantitativamente a cada solución candidata. Estas candidatas pueden ser soluciones que ya se sabe que funcionan, con el objetivo de que el AG las mejore, pero se suelen generar aleatoriamente.

A partir de ahí, AG evalúa cada candidata de acuerdo con la función de aptitud. Por supuesto, se debe tener en cuenta que estas primeras candidatas generadas aleatoriamente, tendrán una eficiencia mínima con respecto a la resolución del problema, y la mayoría no funcionarán en absoluto. Sin embargo, por puro azar, unas pocas pueden ser prometedoras, pudiendo mostrar algunas características que muestren, aunque sólo sea de una forma débil e imperfecta, cierta capacidad de solución del problema.

Estas candidatas prometedoras se conservan y se les permite reproducirse. Se realizan múltiples copias de ellas, pero estas copias no son perfectas, sino que se les introducen algunos cambios aleatorios durante el proceso de copia, a modo de las mutaciones que pueden sufrir los descendientes de una población. Luego, esta descendencia digital prosigue con la siguiente generación, formando un nuevo conjunto de soluciones candidatas, y son de nuevo sometidas a una ronda de evaluación de aptitud. Las candidatas que han empeorado o no han mejorado con los cambios en su código son eliminadas de nuevo; pero, de nuevo, por puro azar, las variaciones aleatorias introducidas en la población pueden haber mejorado a algunos individuos, convirtiéndolos en mejores soluciones del problema, más completas o más eficientes. El proceso se repite las iteraciones que haga falta, hasta que obtengamos soluciones suficientemente buenas para nuestros propósitos.

1.5 DEEP LEARNING

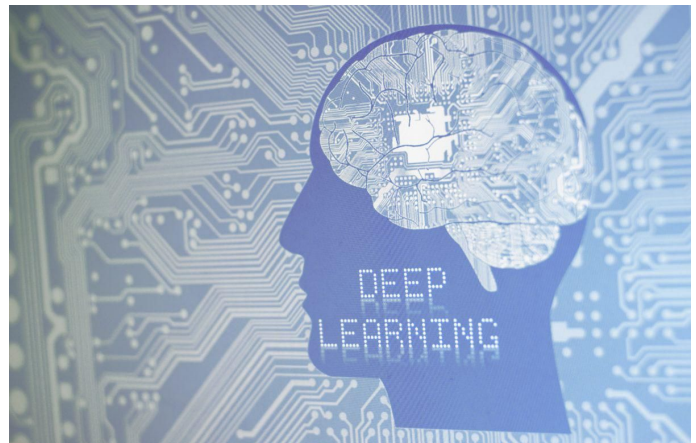
El Deep Learning o aprendizaje profundo se define como un algoritmo automático estructurado o jerárquico que emula el aprendizaje humano con el fin de obtener ciertos conocimientos. Destaca porque no requiere de reglas programadas previamente, sino que el propio sistema es capaz de «aprender» por sí mismo para efectuar una tarea a través de una fase previa de entrenamiento.

A su vez, también se caracteriza por estar compuesto por redes neuronales artificiales entrelazadas para el procesamiento de información. Se emplea principalmente para la automatización de análisis predictivos.

Los algoritmos que componen un sistema de aprendizaje profundo se encuentran en diferentes capas neuronales compuestas por pesos (números). El sistema está dividido principalmente en 3 capas:

- Capa de entrada (Input Layer): Está compuesto por las neuronas que asimilan los datos de entrada, como por ejemplo imagen o una tabla de datos.
- Capa oculta (Hidden Layer): Es la red que realiza el procesamiento de información y hacen los cálculos intermedios. Cada más neurona en esta capa haya, más complejos son los cálculos que se efectúan.
- Salida (Output Layer): Es el último eslabón de la cadena, y es la red que toma la decisión o realiza alguna conclusión aportando datos de salida.

Mientras que los algoritmos tradicionales de aprendizaje automático son lineales, los algoritmos de aprendizaje profundo se apilan en una jerarquía de creciente complejidad y abstracción. Para entender el aprendizaje profundo, imagine a un niño cuya primera palabra es "perro". El niño aprende lo que es (y lo que no es) un perro señalando objetos y diciendo la palabra "perro". El padre dice "Sí, eso es Perro" o "No, eso no es un perro". Mientras el niño continúa apuntando a los objetos, se vuelve más consciente de las características que poseen todos los perros. Lo que el niño hace, sin saberlo, es aclarar una abstracción compleja (el concepto de perro) construyendo una jerarquía en la que cada nivel de abstracción se crea con el conocimiento que se obtuvo de la capa precedente de la jerarquía.



Los programas informáticos que utilizan el aprendizaje profundo pasan por el mismo proceso. Cada algoritmo en la jerarquía aplica una transformación no lineal en su entrada y utiliza lo que aprende para crear un modelo estadístico como salida. Las iteraciones continúan hasta que la salida ha alcanzado un nivel de precisión aceptable. El número de capas de procesamiento a través de las cuales los datos deben pasar es lo que inspiró la etiqueta de profundidad ("deep").

En el aprendizaje tradicional de las máquinas, el proceso de aprendizaje es supervisado y el programador tiene que ser muy, muy específico al decirle a la computadora qué tipos de cosas debe buscar para decidir si una imagen contiene un

perro o no contiene un perro. Este es un proceso laborioso llamado extracción de características y la tasa de éxito de la computadora depende totalmente de la capacidad del programador para definir con precisión un conjunto de características para "perro". La ventaja del aprendizaje profundo es que el programa construye el conjunto de características por sí mismo sin supervisión. Esto no es sólo más rápido, sino que por lo general es más preciso.

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://empresas.blogthinkbig.com/redes-neuronales-artificiales/>

[3]

<https://medium.com/@williamkhepri/redes-neuronales-que-son-a64d022298e0>

[4]

<https://www.tdx.cat/bitstream/handle/10803/6887/0/4Rpp04de11.pdf>

[5]

<https://www.tecnologias-informacion.com/sistemas-expertos.html>

[6]

<https://sites.google.com/site/proyectoInteligenciaArtificial/indice/los-sistemas-expertos>

[7]

<https://medium.com/soldai/algoritmos-gen%C3%A9ticos-en-inteligencia-artificial-parte-i-introducci%C3%B3n-c87b1f52c34f>

[8]

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

[9]

<https://www.smartpanel.com/que-es-deep-learning/>