

Computación Blanda

Soft Computing

Autor: Nestor Augusto Marmolejo Calderon
 IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia
 Correo-e: nestor.marmolejo@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

Una red neuronal es un modelo simplificado e inspirado en las redes neuronales biológicas del cerebro humano que emula el modo en el que este procesa la información. Están constituidas por elementos que se comportan de forma similar a una neurona biológica en sus funciones más comunes. Dichos elementos están organizados de una forma similar a como se presentan en el cerebro.

Al margen de “parecerse” al cerebro presentan una serie de características propias del cerebro. Por ejemplo las ANN aprenden de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos.

Aprender: Adquirir el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Las ANN pueden cambiar su comportamiento en función del entorno. Se les muestra un conjunto de entradas y ellas mismas se ajustan para producir unas salidas consistentes.

Generalizar: extender o ampliar una cosa. Las ANN generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, dentro de un margen, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de ruido o distorsión.

Abstraer: aislar mentalmente o considerar por separado las cualidades de un objeto. Algunas ANN son capaces de abstraer la esencia de un conjunto de entradas que aparentemente no presentan aspectos comunes o relativos.

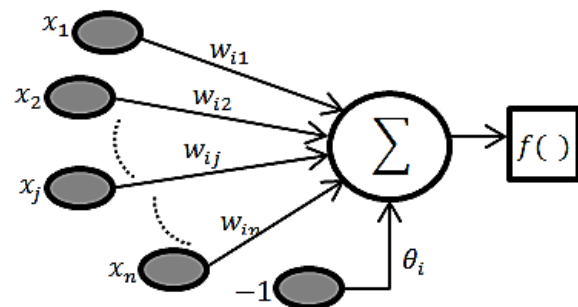


Fig. 1. Diagrama de una neurona artificial (PE)

En las Redes Neuronales Artificiales, ANN, la unidad análoga a la neurona biológica es el elemento procesador, PE (process element). Un elemento procesador tiene varias entradas y las combina, normalmente con una suma básica. La suma de las entradas es modificada por una función de transferencia y el valor de la salida de esta función de transferencia se pasa directamente a la salida del elemento procesador.

La salida del PE se puede conectar a las entradas de otras neuronas artificiales (PE) mediante conexiones ponderadas correspondientes a la eficacia de la sinapsis de las conexiones neuronales.

Siendo una red neuronal un conjunto de unidades elementales PE conectadas de una forma concreta. El interés de las ANN no reside solamente en el modelo del elemento PE sino en las formas en que se conectan estos elementos procesadores. Generalmente los elementos PE están organizados en grupos llamados niveles o capas. Una red típica consiste en una secuencia de capas con conexiones entre capas adyacentes consecutivas.

Existen dos capas con conexiones con el mundo exterior. Una capa de entrada, buffer de entrada, donde se presentan los datos a la red, y una capa buffer de salida que mantiene la respuesta de la red a una entrada. El resto de capas reciben el nombre de capas ocultas.

La red aprende examinando los registros individuales, generando una predicción para cada registro y realizando ajustes a las ponderaciones cuando realiza una predicción incorrecta. Este proceso se repite muchas veces y la red sigue mejorando sus predicciones hasta haber alcanzado uno o varios criterios de parada.

Al principio, todas las ponderaciones son aleatorias y las respuestas que resultan de la red son, posiblemente, disparatadas. La red aprende a través del entrenamiento.

Continuamente se presentan a la red ejemplos para los que se conoce el resultado, y las respuestas que proporciona se comparan con los resultados conocidos. La información procedente de esta comparación se pasa hacia atrás a través de la red, cambiando las ponderaciones gradualmente. A medida que progresa el entrenamiento, la red se va haciendo cada vez más precisa en la replicación de resultados conocidos. Una vez entrenada, la red se puede aplicar a casos futuros en los que se desconoce el resultado.

1.2 LÓGICA DIFUSA

La lógica difusa, es la lógica aplicada a conceptos que pueden tomar un valor cualquiera de veracidad dentro de un conjunto de valores que oscilan entre dos extremos, la verdad absoluta y la falsedad total, lo cual permite tratar información imprecisa, como estatura media o temperatura baja, en términos de conjuntos borrosos o difusos que se

combinan en reglas para definir acciones como: *si la temperatura es alta entonces enfriar mucho*. De esta manera, los sistemas de control basados en lógica difusa combinan variables de entrada, definidas en términos de conjuntos difusos, por medio de grupos de reglas que producen uno o varios valores de salida.

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.

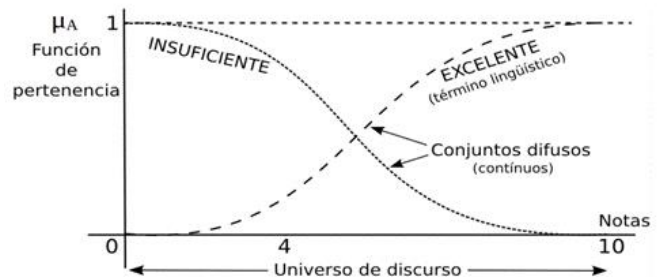


Fig. 2. Representación de los conjuntos difusos: Insuficiente y Excelente

La lógica difusa se utiliza cuando la complejidad del proceso en cuestión es muy alta y no existen modelos matemáticos precisos, para procesos altamente no lineales y cuando se envuelven definiciones y conocimiento no estrictamente definido. En cambio, no es una buena idea usarla cuando algún modelo matemático ya soluciona eficientemente el problema, cuando los problemas son lineales o cuando no tienen solución. Esta técnica se ha empleado con bastante éxito en la industria, principalmente en Japón, y cada vez se está usando en gran multitud de campos. La primera vez que se usó de forma importante fue en el metro japonés, con excelentes resultados.

En Inteligencia artificial, la lógica difusa se utiliza para la resolución de una variedad de problemas, principalmente los relacionados con control de procesos industriales complejos y sistemas de decisión en general, la resolución la comprensión de datos. Los sistemas de lógica difusa están también muy extendidos en la tecnología cotidiana, por ejemplo en cámaras digitales, sistemas de aire acondicionado, etc. Los sistemas basados en lógica difusa imitan la forma en que toman decisiones los humanos, con la ventaja de ser mucho más rápidos. Estos sistemas son generalmente robustos y tolerantes

a imprecisiones y ruidos en los datos de entrada. Consiste en la aplicación de la lógica difusa con la intención de imitar el razonamiento humano en la programación de computadoras. Con la lógica convencional, las computadoras pueden manipular valores estrictamente duales, como verdadero/falso, sí/no o ligado/desligado. En la lógica difusa, se usan modelos matemáticos para representar nociones subjetivas, para valores concretos que puedan ser manipuladas por los ordenadores.

1.3 SISTEMAS EXPERTOS

Los sistemas expertos, son sistemas informáticos que simulan el proceso de aprendizaje, de memorización, de razonamiento, de comunicación y de acción en consecuencia de un experto humano en cualquier rama de la ciencia.

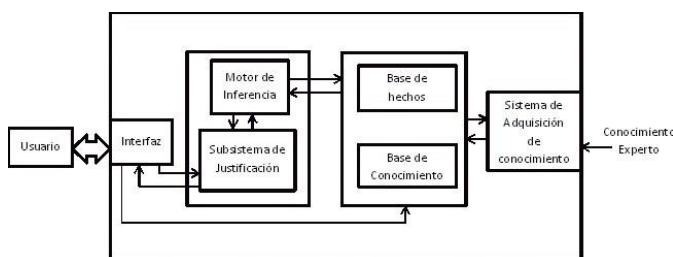


Fig. 3. Estructura de un sistema experto

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.

Esencialmente, cuentan con cinco componentes:

Base de conocimiento: en este componente se representan los hechos y reglas. Aquí se almacena el conocimiento en un dominio particular, así como en las reglas para resolver un problema, procedimientos y datos intrínsecos relevantes para el dominio.

Es el repositorio de los hechos, es decir, un gran contenedor de conocimiento que se obtiene de diferentes expertos de un campo específico.

Motor de inferencia: es el cerebro del sistema experto. Su función es obtener el conocimiento relevante de la base de conocimientos, interpretarlo y encontrar una solución relevante para el problema del usuario. Contiene las reglas de su base de conocimiento y las aplica a los hechos conocidos para inferir nuevos hechos.

Así, proporciona razonamiento sobre la información en la base de conocimiento. Los motores de inferencia también pueden incluir una explicación y habilidades de depuración.

Módulo de adquisición de conocimiento y aprendizaje: es la parte de estos sistemas que permite que el sistema experto adquiera cada vez más conocimiento de diversas fuentes y lo almacene en la base de conocimiento.

Interfaz de usuario: es la parte más crucial del sistema experto. Este módulo hace posible que un usuario no experto interactúe con el sistema experto y encuentre una solución a un problema.

El componente toma la consulta de un usuario en una forma legible y pasa al motor de inferencia. Una vez allí, entran en funcionamiento los otros componentes y posteriormente muestra los resultados al usuario. En otras palabras, es una interfaz que ayuda al usuario a comunicarse con el sistema experto.

Módulo de explicación: en este módulo el sistema experto da una explicación al usuario sobre cómo el sistema experto llegó a una conclusión particular.

De esa manera, argumenta los resultados presentados con la mayor exactitud.

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 ésta 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.

Siendo estos de gran ayuda para 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.

1.4 ALGORITMOS GENÉTICOS

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.

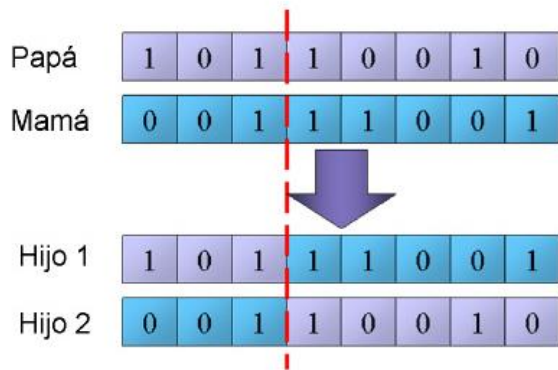


Fig. 4. Intercambio de información genética

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.

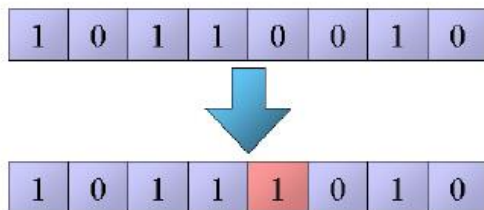


Fig. 5. Una mutación modifica al azar parte del cromosoma de los individuos

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.

La aplicación más común de los algoritmos genéticos ha sido la solución de problemas de optimización, en donde han mostrado ser muy eficientes y confiables. Sin embargo, no todos los problemas pudieran ser apropiados para la técnica, y se recomienda en general tomar en cuenta las siguientes características del mismo antes de intentar usarla:

Su espacio de búsqueda (i.e., sus posibles soluciones) debe estar delimitado dentro de un cierto rango.

Debe poderse definir una función de aptitud que nos indique qué tan buena o mala es una cierta respuesta.

Las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en la computadora.

El primer punto es muy importante, y lo más recomendable es intentar resolver problemas que tengan espacios de búsqueda discretos aunque éstos sean muy grandes. Sin embargo, también podrá intentarse usar la técnica con espacios de búsqueda continuos, pero preferentemente cuando exista un rango de soluciones relativamente pequeño.

La función de aptitud no es más que la función objetivo de nuestro problema de optimización. El algoritmo genético únicamente maximiza, pero la minimización puede realizarse fácilmente utilizando el recíproco de la función que maximiza (debe cuidarse, por supuesto, que el recíproco de la función no genere una división por cero). Una característica que debe tener esta función es que tiene ser capaz de "castigar" a las malas soluciones, y de "premiar" a las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez.

La codificación más común de las soluciones es a través de cadenas binarias, aunque se han utilizado también números reales y letras. El primero de estos esquemas ha gozado de mucha popularidad debido a que es el que propuso originalmente Holland, y además porque resulta muy sencillo de implementar.

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.

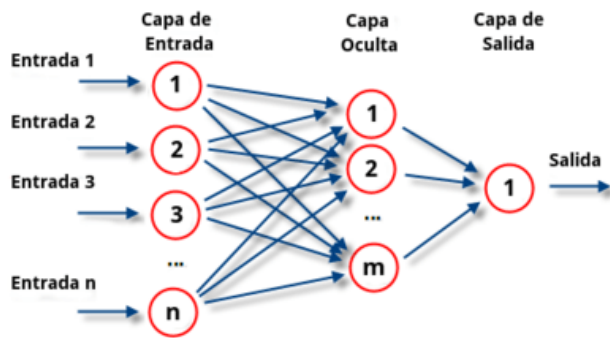


Fig. 6. Arquitectura de una red neural simple

Los algoritmos que componen un sistema de aprendizaje profundo se encuentra 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 neuronas 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.

Hay que mencionar, que las redes neuronales pueden ser virtuales, es decir, creadas en un espacio ficticio dentro de un ordenador como pueden ser los Open Sources Scili Learn, TensorFlow, Keras o Microsoft CNTk, o pueden estar compuestas por silicio, el material más apropiado para crear redes neuronales físicas y tangibles.

Se necesita mucho poder de cómputo para resolver problemas de deep learning debido a la naturaleza iterativa de los algoritmos de deep learning, su complejidad conforma aumenta el número de capas y los grandes volúmenes de datos que se necesitan para entrenar a las redes.

La naturaleza dinámica de los métodos de deep learning – su capacidad de mejorar y adaptarse continuamente a cambios en el patrón de información implícito – presenta una gran oportunidad para introducir un comportamiento más dinámico a la analítica.

Una mayor personalización de la analítica de clientes es una posibilidad. Otra gran oportunidad es mejorar la precisión y el desempeño en aplicaciones donde se han utilizado redes neuronales por largo tiempo. A través de mejores algoritmos y más poder de cómputo podemos agregar mayor profundidad.

Aunque el enfoque actual del mercado de las técnicas de deep learning está en aplicaciones de cómputo cognitivo, también hay un gran potencial en aplicaciones analíticas más tradicionales; por ejemplo, el análisis de series de tiempo.

Otra oportunidad es simplemente ser más eficiente y simplificado en operaciones analíticas existentes. Recientemente, SAS experimentó con redes neuronales profundas en problemas de transcripción de habla a texto. Comparado con las técnicas estándares, el índice de errores en palabras disminuyó más de 10% cuando se aplicaron redes neuronales profundas. También eliminaron cerca de 10 pasos del procesamiento de datos, ingeniería de características y modelado. Los impresionantes incrementos de desempeño y los ahorros de tiempo cuando se comparan con la ingeniería de características se traducen en un cambio de paradigma.



Nestor Augusto Marmolejo Calderon

(S'20) Nacido en Pereira-Risaralda Colombia el 27 de Agosto de 1999, bachiller del Instituto Técnico Superior, Técnico en implementación y mantenimiento de equipos electrónicos industriales y estudiante de ingeniería en sistemas y computación en la Universidad Tecnológica de Pereira, cursando

actualmente 8avo semestre, becado por el programa de la alcaldía de Pereira llamado 'Universidad para cuba'

REFERENCIAS

Referencias en la Web:

- [1] <https://computerhoy.com/reportajes/tecnologia/inteligencia-artificial-469917>
- [2] https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html
- [3] <https://medium.com/@williamkhepri/redes-neuronales-que-son-a64d022298e0>
- [4] <https://www.tdx.cat/bitstream/handle/10803/6887/0/4Rpp04de11.pdf>
- [5] <http://www.cs.us.es/~fsancho/?e=97>

[6]

https://www.ecured.cu/L%C3%B3gica_difusa

[7]

https://www.ecured.cu/Sistemas_expertos

[8]

https://www.palermo.edu/ingenieria/pdf2014/13/CyT_13_24.pdf

[9]

https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-50062012000100003

[10]

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

[11]

<http://conogasi.org/articulos/algoritmos-geneticos/>

[12]

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

[13]

https://www.sas.com/es_co/insights/analytics/deep-learning.html

[14]

https://www.ecured.cu/Algoritmos_Gen%C3%A9ticos