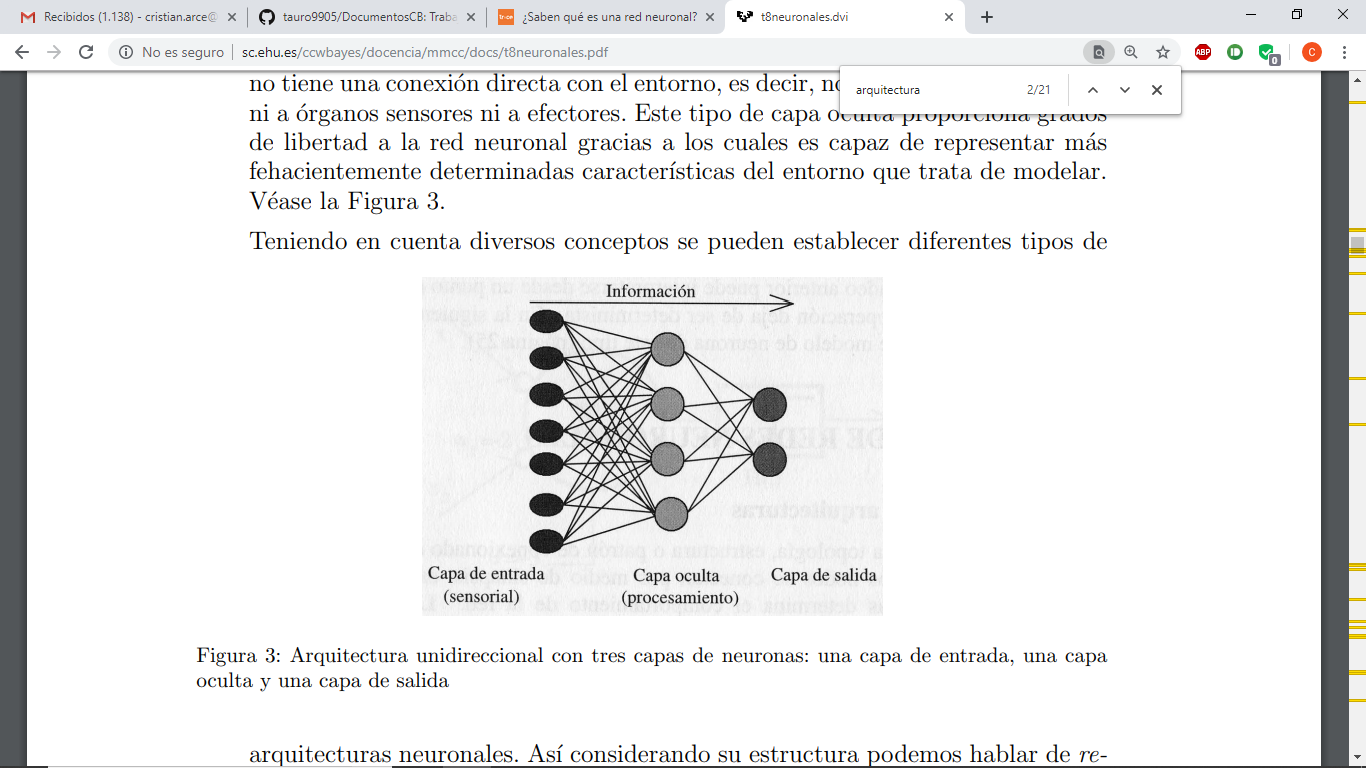
TALLER N°1 – COMPUTACION BLANDA

1. Hacer descripción, arquitectura, ejemplos de uso y herramientas disponibles
2. Redes neuronales, lógica difusa, sistemas expertos, algoritmos genéticos, agente inteligente
3. Al final del documento hacer una lista de problemas interesantes

**REDES NEURONALES.**

* Descripción: La **red neuronal** es similar a las neuronas que tenemos los seres humanos y que están conectadas a nuestro cerebro, pero llevada al caso artificial, son redes que se conectan entre sí mediante pequeños programas que identifican **patrones** en los datos de forma colaborativa, lo que les permite **procesar**, analizar e incluso **aprender**.
* Arquitectura: Se denomina arquitectura a la topología, estructura o patrón de conexionado de una red neuronal. En una red neuronal artificial los nodos se conectan por medio de sinapsis, estando el comportamiento de la red determinado por la estructura de conexiones sinápticas. Estas conexiones sinápticas son direccionales, es decir, la información solamente puede propagarse en un único sentido (desde la neurona presináptica a la postsináptica). En general las neuronas se suelen agrupar en unidades estructurales que denominaremos capas. El conjunto de una o más capas constituye la red neuronal. Se distinguen tres tipos de capas: de entrada, de salida y ocultas. Una capa de entrada, también denominada sensorial.



* Ejemplos de uso: En el mundo de los negocios la aportación de las**redes neuronales artificiales,** ya sea a nivel estratégico, táctico u operativo, es extremadamente valiosa, precisamente por su importante contribución a la analítica predictiva que se pone de manifiesto:

En la construcción de modelos explicativos, que ayudan a explorar conjuntos de datos en busca de variables o grupos de variables pertinentes.

En su capacidad para aproximar cualquier función continua, haciendo posible que el analista no necesite tener ninguna hipótesis sobre el modelo subyacente.

El valor clave de las**redes neuronales artificiales** es su capacidad para modelar funciones no lineales muy complejas, aunque también se adaptan bien a problemas altamente dimensionales, donde el número potencial de predictores es muy grande.

* Herramientas disponibles: TensorFlow, Caffe, Caffe 2 y ONNX.

**LOGICA DIFUSA.**

* Descripción: 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. Así, por ejemplo, una persona que mida 2 metros es claramente una persona alta, si previamente se ha tomado el valor de persona baja y se ha establecido en 1 metro. Ambos valores están contextualizados a personas y referidos a una medida métrica línea.
* Arquitectura: Para cada conjunto difuso, existe asociada una función de pertenencia para sus elementos, que indican en qué medida el elemento forma parte de ese conjunto difuso. Las formas de las funciones de pertenencia más típicas son trapezoidal, lineal y curva.

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.
* Ejemplos de uso:
  + Sistemas de control de acondicionadores de aire
  + Sistemas de foco automático en cámaras fotográficas
  + Optimización de sistemas de control industriales
  + Sistemas de reconocimiento de escritura
  + Mejora en la eficiencia del uso de combustible en motores
  + Sistemas expertos del conocimiento
  + Tecnología informática
  + Bases de datos difusas: Almacenar y consultar información imprecisa. Para este punto, por ejemplo, existe el lenguaje FSQL.
* Herramientas:

|  |
| --- |
| * + fuzzyTECH: software development tools for fuzzy logic |
| * + jFuzzyLogic: Open Source Fuzzy Logic |
| * + mbFuzzIT: a package of Java classes for Fuzzy inference |
| * + Metarule: rule based language for expressing knowledge |
| * + SINE: Simple INference Engine |
| * + Source code on Neural Networks and Fuzzy Systems by the Research group NNandFS (Magdeburg). |
| * + Type-2 Fuzzy Logic Software |
| * + XFuzzy: Fuzzy system development environment. |

**SISTEMAS EXPERTOS.**

* Descripción: Un Sistema Experto es aquel capaz de almacenar el conocimiento de un experto en una especialidad determinada y limitada, y a su vez de solucionar problemas mediante la inducción- deducción lógica.  
  Los sistemas expertos son programas de computación que capturan el conocimiento de un experto y tratan de imitar su proceso de razonamiento cuando resuelven los problemas en un determinado dominio.
* Arquitectura: Un **sistema experto se compone de**:
  + Base de hechos
  + Base de conocimientos
  + Motor de inferencia
  + Módulos de comunicación o, de entrada-salida que se subdivide en:
  + Módulo de consulta o del usuario
  + Módulo de trabajo o del experto



* Ejemplos de uso: Los sistemas expertos están implementados en proyectos como, MYCIN, es un sistema experto para la realización de diagnósticos en el área de la medicina; XCON, es un sistema experto para configuraciones, desarrollado por la Digital Equipment Corporation y DELTA, es un sistema experto que ayuda a los médicos en el diagnósticos y reparación de trenes Diesel eléctricas.
* Herramientas:
  + Python
  + Java
  + Prolog
  + Jess
  + Winprolog
  + Expert System Builder
  + G2.

**ALGORITMOS GENETICOS.**

* Descripción: 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 de acorde con los principios de la selección natural y la supervivencia de los más fuertes.

**ASUNTO CBG01**