Interfaz de programación de aplicaciones e interfaz gráfica para modelos de memorias asociativas.

Nombre del alumno: Oscar Adrian Mariscal Arredondo

Asesor: Israel Román Godínez.

Carrera: INNI

Para validación de los módulos:

* Módulo 1
* Módulo 2
* Módulo 4

# Resumen

Desde los inicios de las ciencias computacionales ha sido muy común que se intente modelar ciertos comportamientos y características de los seres humanos, una de ellas es la capacidad de aprender. La rama de las ciencias que se encarga de intentar modelar dicha característica es la inteligencia artificial. Los modelos de aprendizaje más comúnmente usados son las redes neuronales y memorias asociativas. Algunas de las tareas en las que se aplican las memorias asociativas son los relacionados con la calisficación.

El presente proyecto modular tiene como objetivo desarrollar una Interfaz de Programación de Aplicaciones (API, por sus siglas en ingles), utilizando el paradigma orientado a objetos, que permita a los investigadores y tecnólogos construir componentes de software para el reconocimiento de patrones, utilizando algoritmos de aprendizaje automático de computadoras (machine learning). Además, con el fin de que esta herramienta pueda ser utilizada por profesionales sin conocimientos de programación, se desarrollará una interfaz gráfica de usuario (GUI, por sus siglas en ingles) que utilice el API antes mencionada. La interfaz gráfica de usuario permitirá llevar a cabo el entrenamiento de la memoria asociativa y permitirá llevara cabo clasificación sobre un conjunto de datos de prueba.

# Antecedentes

En la ciencias de la computación se han desarrollado modelos que tratan de asemejar la habilidad de aprender del ser humano, uno de estos son las redes neuronales artificiales, las cuales son modelos que tratan de asemejar las funciones y estructuras biológicas de las redes neuronales. Éstas cuentan con un subtema llamado memorias asociativas las cuales, a su vez, cuentan con varios algoritmos, por ejemplo LearnMatrix de Steinbuch [1,2] correlograph de willshaw, buneman & longuet-Higgins [2], linear associator de Anderson-kohonen [4,5], Memoria asociativa de hopfield [6], modelos como ADAM (Advanced Distributed Associative Memory) de Austin [7], BAM (Bidirectional Associative Memory) de kosko [8], SDM (Sparse Distributed Memory) de Kanerva [9], Memorias Morfológicas de Ritter, Diaz-de-león & Sussner[10] y las Memorias asociativas Alfa-Beta de Yañez-Márquez [11]. En esto proyecto vamos a tratar en específico con el modelo de learnMatrix de steinbuch.

La lernmatrix es una memoria **M** que relaciona patrones de entrada y salida, los cuales se encuentran representados por vectores columna, **x** y **y** respectivamente. Cada uno de los patrones de entrada y salida se encuentran relacionados respectivamente, siendo similar a una pareja ordenada (**x,y**).La memoria asociativa de Steinbuch tiene dos fases la primera siendo una fase de aprendizaje y la segunda siendo una de recuperación (reconocimiento de patrones).

Por otro lado, una plataforma de cliente enriquecido (RCP por sus siglas en inglés) es una herramienta de programación que hace más fácil la integración de componentes de software independientes. Con una RCP, los programadores pueden construir sus propias aplicaciones en plataformas informáticas existentes. En lugar de tener que escribir una aplicación completa desde cero, pueden beneficiarse de características probadas y comprobadas del marco proporcionado por la plataforma.

En una búsqueda realizada en uno de los repositorios más conocidos de algoritmos de inteligencia artificial Weka [12] no se encontró ninguna aplicación del algoritmo de memoria asociativa de steinbuch. Por lo que se detecta la necesidad de implementar dichos modelos.

# Justificación

Actualmente, el conocimiento teórico sobre el modelo de memorias asociativas se encuentra esparcido o inaccesible, por lo que encontrar herramientas computacionales que cuenten con implementaciones de dicho modelos es complicado.

Este proyecto busca diseñar e implementar un entorno de trabajo en dos niveles diferentes: 1) nivel *intermedio* orientado a desarrolladores de aplicaciones del área de machine learning; y 2) nivel *alto*, una interfaz gráfica de usuario para las personas que deseen utilizar dicho modelo sin necesidad de programarlo. Además de de ser una alternativa para las diferentes aplicaciones que implementan algoritmos de aprendizaje máquina para la clasificación.

# Objetivos y metas

## Objetivos generales

Diseñar e implementar el modelo de memorias asociativa Learnmatrx a través de una API[13], que va a dirigida a investigadores y tecnólogos del área de las ciencias computacionales; y por otro lado una GUI en Eclipse RCP, con el fin de poner a disposición de los investigadores o estudiantes del tema de aprendizaje automático de computadoras.

## Objetivos específicos

Los objetivos específicos.

* Diseño, utilizando el paradigma orientado a objetos, del modelo de memoria asociativa learnmatrix.
* Implementación de Interfaz gráfica usando el modelo de Eclipse RCP.
* Crear la documentación para el diseño como:
  + Requerimientos del software
  + Diagrama de clases
  + Definición de casos de usos
  + Manual de usuario

## Productos

Los productos que se esperan obtener son:

* Un API sobre los algoritmos que se van a implementar
* Una interfaz gráfica que acompañara la API
* Documentación del diseño como:
  + Requerimientos del software
  + Diagrama de clases
  + Definición de casos de usos
  + Manual de usuario

# Metodología

* Documentarse sobre el algoritmo de LearnMatrix.
* Documentarse sobre la forma de trabajo en eclipse RCP.
* Levantamiento de requerimientos para la API.
* Levantamiento de requerimientos para la GUI.
* Diseñar la Interfaz de programación de aplicaciones (API).
* Diseñar la Interfaz gráfica de usuarios (GUI).
* Implementación de la API.
* Implementación de la GUI.
* Realizar la documentación tanto de la API como de la GUI.

# Prototipo

El prototipo de esta aplicación consistirá de una API sobre el algoritmo de la learnmatrix de steinbuch, además de una interfaz gráfica usando ecliplse RCP que implementa la API antes mencionada.

# Referencias

[1] Steinbuch, K., & Frank, H. (1961). Nichtdigitale lernmatrizen als perzeptoren.*Biological Cybernetics*, *1*(3), 117-124.

[2] Steinbuch, Karl. "Die lernmatrix." *Biological Cybernetics* 1.1 (1961): 36-45.

[3] Willshaw, D. J., Buneman, O. P., & Longuet-Higgins, H. C. (1969). Non-holographic associative memory. *Nature*.

[4] Anderson, J. A. (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences*, *14*(3), 197-220.

[5] Kohonen, T. (1972). Correlation matrix memories. *Computers, IEEE Transactions on*, *100*(4), 353-359.

[6] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554-2558.

[7] Austin, J. (1987, June). ADAM: A distributed associative memory for scene analysis. In *Proceedings of First International Conference on Neural Networks*(Vol. 4, p. 285).

[8] Kosko, B. (1988). Bidirectional associative memories. *Systems, Man and Cybernetics, IEEE Transactions on*, *18*(1), 49-60.

[9] Kanerva, P. (1988). *Sparse distributed memory*. MIT press.

[10] Ritter, Gerhard X., Peter Sussner, and J. L. Diza-de-Leon. "Morphological associative memories." *Neural Networks, IEEE Transactions on* 9.2 (1998): 281-293.

[11] Yáñez Márquez, C. (2003). Memorias Asociativas Basadas en Relaciones de Orden y Operaciones Binarios. *Computación y Sistemas*, *6*(004).

[12] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10-18.

[13]Reddy, M. (2011). *API Design for C++*. Elsevier.