



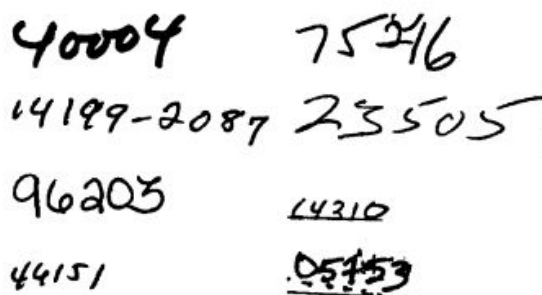
## Laboratorio 1

18/04/2019

### Fundamento teórico

Presentado por Frank Rosenblatt en 1958, y basado en el previo modelo de McCulloch y Pitts, el perceptrón simple es un modelo matemático que representa la unidad celular básica presente en la actividad neuronal del ser humano. Con la capacidad de reaccionar ante un estímulo que logre superar su umbral de excitación y entregar un resultado a dicho estímulo, el perceptrón simple es la base para la implementación de herramientas de inteligencia computacional como las redes neuronales.

En la vida cotidiana esta herramienta ha tenido diversos usos para resolver problemas que necesitan de una gran capacidad de procesamiento de datos, un ejemplo es el reconocimiento de los dígitos manuscritos escritos en el código postal (LeCun, 1989).



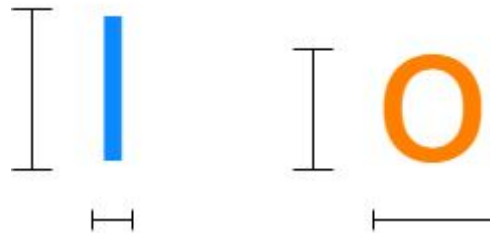
Handwritten postal codes (MNIST dataset) shown in the figure:

40004	75216
14189-2087	23505
96203	14310
44151	05453

*Figura 1: Muestra de códigos postales (LeCun, 1989).*

Si bien para lograr el reconocimiento de éstos dígitos fue necesaria una red neuronal profunda especialmente construida para esta tarea, podemos entender el proceso de reconocimiento como un aprendizaje de máquina supervisado que logra extraer las características o patrones que identifiquen a cada dígito y ajuste la relevancia de cada una para así refinar la capacidad de clasificación de esta red.

Simplificando este ejercicio podríamos decir que es posible clasificar un caracter según dos de sus características: alto y ancho.



*Figura 2: Características distintivas propuestas de una letra.*

### Actividades

1. Implementar un perceptrón simple en Python utilizando Jupyter Notebook. Este debe entregar una salida respecto a una entrada según su función de activación y ajustar sus pesos en cada iteración según corresponda.
2. Probar el perceptrón implementado para una compuerta lógica AND, OR y XOR con una entrada de tamaño 4.
3. Probar el perceptrón implementado para clasificar las entradas del dataset "letters.csv" determinando a qué letra corresponde cada entrada según las características recibidas.
4. Calcular el error obtenido durante el entrenamiento del perceptrón y expresarlo a través de un gráfico.
5. Analizar los resultados obtenidos y concluir sobre los resultados obtenidos en las actividades 2 y 3.

### Entregable

Archivo .ipynb en un .zip que contenga el código de las actividades mencionadas en el punto anterior.

**Fecha de entrega: 3 de mayo del 2019.**

### Fuentes:

Anaconda:

<https://www.anaconda.com/distribution/>

Jupyter:

<https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>