



UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA
FUNDAMENTOS DE APRENDIZAJE PROFUNDO CON REDES NEURONALES



Laboratorio 1

09/09/2019

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 en el código postal de la correspondencia (LeCun, 1989).

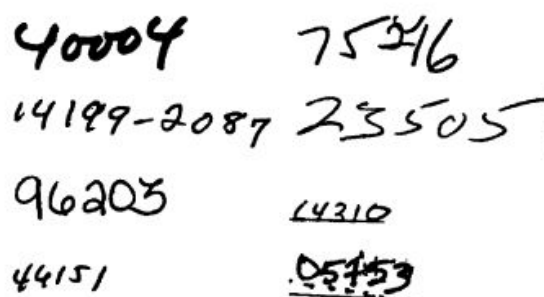


Figura 1: Muestra de códigos postales (LeCun, 1989).

Si bien para lograr el reconocimiento de éstos dígitos fue necesario utilizar 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 de ellas 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.

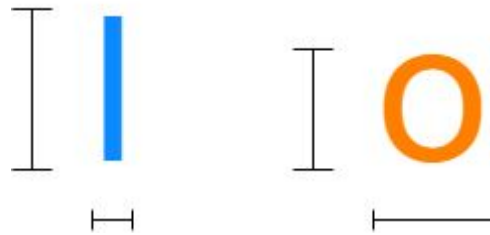


Figura 2: Características distintivas propuestas de una letra.

Actividades

1. Implementar un perceptrón simple genérico en Python utilizando Jupyter Notebook (versión 2.7 o 3.1) (Google Colab, Anaconda, etc). El perceptrón implementado debe tener una función de activación, ajustar los pesos en cada iteración según corresponda y entregar una salida acorde a la entrada que reciba.
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 en las actividades 2 y 3 durante el entrenamiento del perceptrón y expresarlo a través de un gráfico Iteración vs error.
5. Analizar y concluir sobre los resultados obtenidos en las actividades 2, 3 y 4.

Entregable

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

Fecha límite de entrega: 23 de septiembre del 2019 hasta las 23:55.

Entornos:

Google Colab:

<https://colab.research.google.com/>

Anaconda:

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

Jupyter:

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