

# Reconocimiento de letras manuscritas con redes neuronales artificiales

## Tercera y cuarta entrega

Edson Raul Cepeda Marquez  
Sonia Alejandra Treviño Rivera

1 de junio de 2020

## 1. Resumen

Este proyecto consiste en una aplicación web que implementa una red neuronal artificial capaz de reconocer letras manuscritas. La aplicación web en cuestión está desarrollada con tres tecnologías que se conectan y permiten su uso:

1. **Angular:** Un framework de código libre desarrollado por Google que facilita el desarrollo de aplicaciones web de una sola página (SPA: Single Page Application). Tiene la ventaja de separar completamente el frontend y el backend lo que permite seguir fácilmente el patrón de diseño MVC (Modelo-Vista-Controlador).
2. **Express:** Un framework de desarrollo web que utiliza el entorno de ejecución Node JS y facilita la creación de REST APIs.
3. **Python:** Un lenguaje de programación interpretado orientado a objetos. Tiene la ventaja de ser simple, lo que permite escribir código legible. Tiene una amplia comunidad de desarrollo y una gran cantidad de librerías para todo tipo de tareas.

En la figura 1 se muestra un diagrama que representa el flujo de información entre las tres tecnologías. Se puede observar que la información fluye en ambas direcciones.

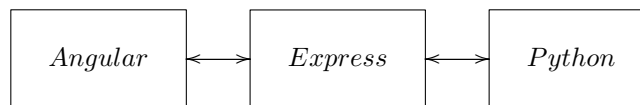


Figura 1: Flujo de información de la aplicación.

La aplicación web además de contener este proyecto, tiene otras secciones asignadas para proyectos futuros. Para este proyecto la sección importante es “Imchar”. Esta sección de la aplicación web contiene una cuadrícula con la que se puede interactuar a través del ratón.

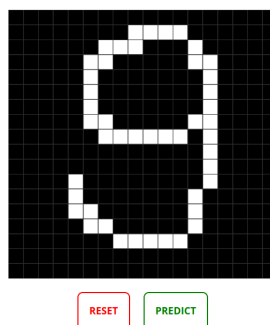


Figura 2: Cuadrícula de la aplicación.

Esta es la puerta para el uso de la red neuronal artificial. En esta cuadrícula es posible dibujar letras. Al dibujar una letra en la cuadrícula y presionar el botón de “Predeict” la aplicación web realiza una petición al servidor de la REST API de express, la cual recibe el estado de la cuadrícula y sus dimensiones. Posteriormente el servidor de express ejecuta un proceso asincrono de python el cual ejecuta un script que lleva a cabo una transformación de los datos enviados en la petición. Por ultimo los datos transformados de la cuadrícula son procesados por la red neuronal previamente entrenada, y devuelve una respuesta al frontend convertida en el caracter que la red neuronal predice (Figura 3).



Figura 3: Respuesta de la red neuronal, también incluye el botón “Reset” para borrar la cuadrícula.

**Para el desarrollo de la red neuronal artificial no se utilizo ninguna librería, todo código correspondiente al algoritmo de aprendizaje fue escrito desde cero.**

Así mismo la red neuronal que procesa la información fue entrenada con un conjunto de datos creado desde cero, para esto se implemento una ruta oculta (figura 4) en la aplicación web en la que es posible utilizar la cuadrícula para generar un conjunto de entranamiento.

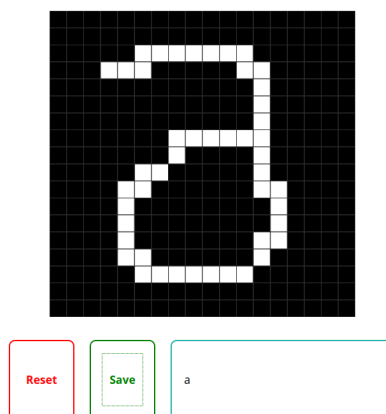


Figura 4: Ruta oculta para la generación de datos de entrenamiento

Los resultados de entrenar la red neuronal con estos datos son que, para las 26 letras del abecedario en ingles por lo menos 21 son predichas correctamente. Las letras que presentan más problemas para ser predichas correctamente son aquellas que comparten cierta similaridad. La red neuronal fue entrenada multiples veces para poder llegar a una configuración de hiperparametros con un mejor rendimiento.

## 2. Introducción

El problema que se intenta resolver al diseñar este sistema, es el de reconocer texto manuscrito para su digitalización. Se diseña una red neuronal convolucional que es entrenada