ABECEDARIO ESPAÑOL

Jose Pedro Zarate Espinosa

April 5, 2021

Abstract

This document implements the spanish alphabet.

1 Introducción

El proyecto que se realizado consta del abecedario español representando cada letra en una matriz de 13 (columnas) * 13 (filas), dando como resultado 169 datos en cada matriz, de acuerdo con la documentación es necesario multiplicar por 0.15 para obtener la cantidad de precisión de identificar cada matriz, es decir que 169 * 0.15 nos da un total de 25.25 patrones a reconocer, los 25 patrones son suficientes para reconocer una buena cantidad de letras del abecedario que vamos a representar debido a que el abecedario consta tan solo de de 27 letras.

El reconocimiento de patrones se llevará acabo por medio del lenguaje c++ haciendo uso de matemáticas aplicada a matrices para poder hacer la función hopfield y calcular los patrones que más se parezcan cuando se le asigne un archivo y pueda compararlos con los archivos de entrenamiento de la red neuronal.

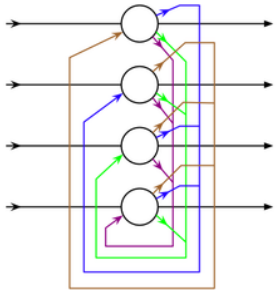

Figure 1: Hopfield Model Neural Network

2 Desarrollo

Para el desarrollo del proyecto se utilizará Octave como sustituto de Matlab, este se usará para el prototipado de la red neuronal la cual después de verificar que funcione en dicho programa se realizará la implementación en código en el lenguaje C++.

```
ı0Matriz =

   1   1   1  -1  -1  -1  -1  -1  -1  -1  -1   1  -1
   1   1   1  -1  -1  -1  -1  -1  -1  -1   1   1  -1
   1   1   1  -1  -1  -1  -1  -1  -1   1   1   1  -1
   1   1   1  -1  -1  -1  -1  -1   1   1   1  -1  -1
   1   1   1  -1  -1  -1  -1   1   1   1  -1  -1  -1
   1   1   1   1   1   1   1   1   1  -1  -1  -1  -1
   1   1   1   1   1   1   1   1  -1  -1  -1  -1  -1
   1   1   1   1   1   1   1   1   1  -1  -1  -1  -1
   1   1   1  -1  -1  -1  -1   1   1   1  -1  -1  -1
   1   1   1  -1  -1  -1  -1  -1   1   1   1  -1  -1
   1   1   1  -1  -1  -1  -1  -1  -1   1   1   1  -1
   1   1   1  -1  -1  -1  -1  -1  -1  -1   1   1  -1
   1   1   1  -1  -1  -1  -1  -1  -1  -1  -1   1  -1

ılast0Matriz =

   1   1   1   0   0   0   0   0   0   0   0   1   0
   1   1   1   0   0   0   0   0   0   0   1   1   0
   1   1   1   0   0   0   0   0   0   1   1   1   0
   1   1   1   0   0   0   0   0   1   1   1   0   0
   1   1   1   0   0   0   0   1   1   1   0   0   0
   1   1   1   1   1   1   1   1   1   0   0   0   0
   1   1   1   1   1   1   1   1   0   0   0   0   0
   1   1   1   1   1   1   1   1   1   0   0   0   0
   1   1   1   0   0   0   0   1   1   1   0   0   0
   1   1   1   0   0   0   0   0   1   1   1   0   0
   1   1   1   0   0   0   0   0   0   1   1   1   0
   1   1   1   0   0   0   0   0   0   0   1   1   0
   1   1   1   0   0   0   0   0   0   0   0   1   0
```

Figure 2: Result Hopfield Model Neural Network in Octave

Una vez teniendo el prototipado en octave, lo siguiente será implementarlo a código,para la implementación se fue probando uno por uno debido que al intentar cargar todos los archivos el programa no lograba reconocer el patrón que contiene cada letra, por ende, se fue agregando al valor de las x1,x2,x3... hasta llegar llegar a x10 debido a los mismos problemas que se presentaron, incluso durante la implementación la letra "J" fue incapaz de reconocer por la red

neuronal ya que no se imprimía de forma correcta aun pasando toda la matriz igual que al archivo que debía comparar.

Durante la implementación de las matrices de entrenamiento se tuvo que hacer pequeños ajustes a los valores con los que se iba a entrenar la red ya que en algunas ocasiones como se mencionó anteriormente la red se confundía y no daba los resultados esperados.
A continuación se muestra un ejemplo de como se tenia pensadas las matrices de entrenamiento para la red.

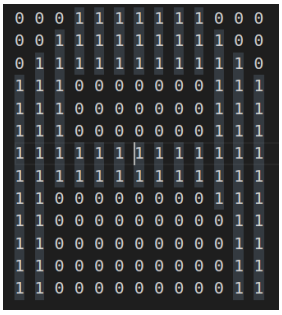

Figure 3: Ejemplo de matriz de entrenamiento inicialmente esperado

El modelo mostrado anteriormente muestra que las letra abarca toda el área posible de la matriz , ya que pensaba que esto traería mejores resultados cuando se le introdujera las demás matrices de cada letra del abecedario, pero al probar una por una se tuvieron que hacer pequeñas modificaciones como las siguientes :

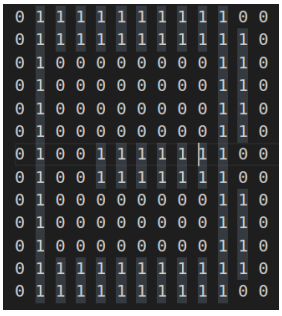

Figure 4: Ejemplo de matriz modificada

Cuando se terminó de hacer las modificaciones a cada archivo de entrenamiento ser verificó que la red funcionara correctamente con cada uno de los 10 archivos que se lograron introducir.

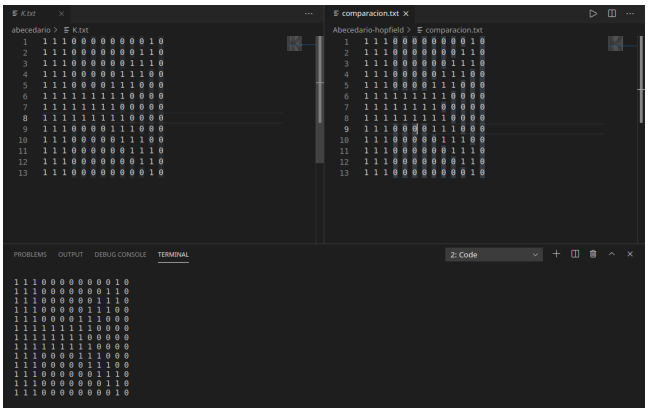

Figure 5: Verificacón de resultados

3 Conclusión

Como conclusión podría decir que no fue completamente los resultados que yo esperaba debido a que los modelos que quería usar para el entrenamiento de la red no funcionaron como lo deseaba, también la letra J no fue capaz de reconocerla aun que modificara la posición y el grosor de la letra, sin embargo,al final con muchas horas de trabajo e intentado entender como era el funcionamiento de la red logré hacer que funcionara para 10 datos de entrenamientos y probar cada uno y modificarlo hasta que la red no se confundiera y me entregara la letra esperada como salida,también por los mismos problemas de modificar cada letra me fue imposible agregar todo el abecedario completo ya que la red comenzaba a colapsar y confundir los valores entregando salidas no existentes de los archivos de entrenamiento.