

Universidad Veracruzana

FACULTAD DE NEGOCIOS Y TECNOLOGÍAS

REPORTE TÉCNICO

Paradigmas de Programación

Autor:

Fhatima Reyes Alejandre

Jueves 17 de Marzo 2021

0.1 Introducción

Un paradigma de programación consiste en uno o varios enfoques y técnicas fundamentales para la programación de software. Es indispensable tener en cuenta esta definición previo al enfoque de este documento, donde se hablará a detalle sobre un tipo de comunicación en el cual, el objetivo es modelar, codificar y ejemplificar a gran detalle el reconocimiento neuronal mediante patrones que nos brindan las Redes Neuronales Artificiales o mejor conocidas como RNA. Este reportaje consiste en centrar una parte teórica lo más clara posible para comprender la parte práctica a través de la arquitectura y algoritmo que puede presentarse para corroborar su factibilidad con una solución que dependa de la situación en la que será implementada.

0.2 Un enfoque hacia RNA y Hopfield

La Neurodinámica se refiere al estudio de RNA vistas como sistemas dinámicos no lineales, dando énfasis en el problema de estabilidad. Las redes neuronales están basadas en el funcionamiento de la neurona biológica residente en el sistema nervioso central, sus orígenes se remontan a los primeros años de la informática, de manera contemporánea a la teoría de la computación de Turing. Una red neuronal artificial es un modelo computacional inspirado en redes neuronales biológicas que puede ser considerada como un sistema de procesamiento de información con características como aprendizaje a través de ejemplos, adaptabilidad, robustez, capacidad de generalización y tolerancia a fallos. La red neuronal artificial puede ser definida como una estructura distribuida, de procesamiento paralelo, formada de neuronas artificiales, llamadas también elementos de procesamiento, interconectados por un gran número de conexiones, las cuales son usadas para almacenar conocimiento.

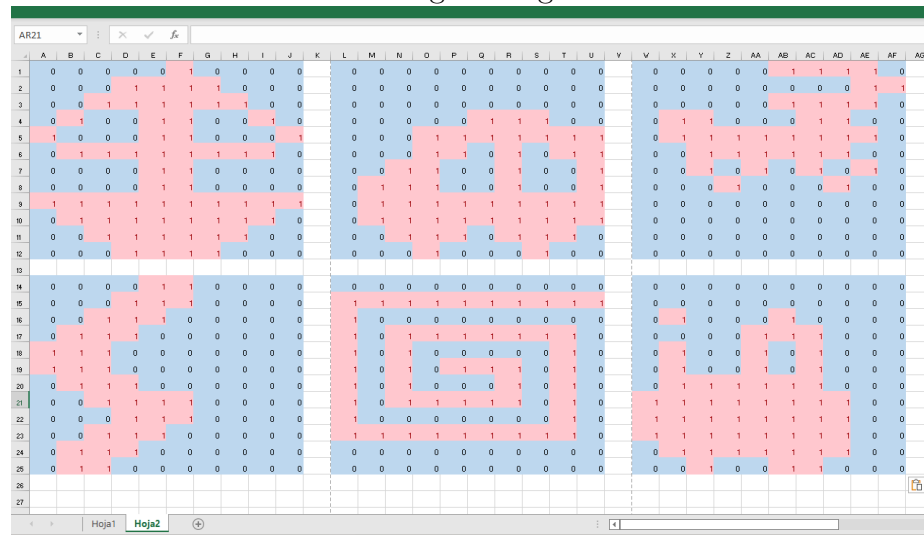
Jhon Hopfield, gracias al trabajo sobre neurofisiología en invertebrados, desarrolló un tipo de red neuronal autoasociativa. La red de Hopfield es una de las redes neuronales artificiales más importantes y ha influido en el desarrollo de multitud de redes posteriores. Hopfield es un modelo de red con el número suficiente de simplificaciones como para extraer analíticamente información sobre las características relevantes del sistema, conservando las ideas fundamentales de las redes construidas en el pasado y presentando una serie de funciones básicas de los sistemas neuronales biológicos.

Para este proyecto y con el objetivo de representar el tema abordado, mi variable a reconocer en el sistema de red neuronal serán los símbolos de transportes (barco, auto, tren, motocicleta) que serán impresas en una matriz de (10 x 12) con el fin de que el algoritmo sea capaz de detectar dicha simbología y pueda notificarnos que ha logrado su cometido mediante un mensaje que contenga dicha figura. La red neuronal de Hopfield es una de las más utilizadas debido principalmente a la facilidad de implementación física utilizando tecnología de integración de gran escala. tenemos que poner en practica la formula que nos especifica la probabilidad de la cantidad de información que pueden almacenar las neuronas del código.

$$(n)neuronas(.15) \quad (1)$$

Con esta formula podemos centrar la probabilidad de neruronas se encuentran en funcionamiento y que pueden retener la información de cada matriz dentro del código. el objetivo es que cada matriz dada pueda representar las figuras deseadas, cada una de ellas con un grosor distinto dependiendo de su tamaño y de lo que representa, todo centrado correctamente en la posición correspondiente dentro de la matriz. Todo esto con el fin de no hacer repetitivo ningun patron y de que nos brinde el numero exacto de iteraciones igual a 1 para así poder corroborar que la matriz ha quedado fielmente plasmada en el scanner de salida. La siguiente representación nos muestra las figuras que se elaboraron en una hoja de calculo de excel para darle forma a matrices.

Tomamos como referencia el siguiente grafico de matrices:



[illegible]

```

Resultado encontrado:

      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

```

```
Resultado ingresado:
```

```
Matrices recorridas hasta hallar el resultado: 1
>> |
```

Ventana de comandos Editor de variables Editor

de que no debía existir algun trazo repetitivo entre cada una de las matrices y si llegaba a haber un percance con el resultado de la figura, era necesario aumentar el grosor de cada una de las figuras o bien, posicionarlas en una dirección diferente con el objetivo de que ninguna de ellas coincidiera en los trazos. Así mismo también nos fue de mucha utilidad un comando que perfecciono el estado de cada una de las matrices. La palabra eye designa a la matriz identidad. Se llevó a cabo la operación de multiplicar la magnitud de la matriz por su transpuesta y a todo ello se le restaba su matriz identidad nuevamente multiplicada por la magnitud.

```

18
19 barmul= bar(:) * bar(:)' - eye(120);
20 barmul
21
22 carmul= car(:) * car(:)' - eye(120);
23 carmul
24
25 trenmul= tren(:) * tren(:)' - eye(120);
26 trenmul
27
28 motomul= moto(:) * moto(:)' - eye(120);
29 motomul
30
31 ovnimul= ovni(:) * ovni(:)' - eye(120);
32 ovnimul
33
34 espmul= esp(:) * esp(:)' - eye(120);
35 espmul
36
37 rayomul= rayo(:) * rayo(:)' - eye(120);
38 rayomul
39
40 %suma de figuras
41
42 w = barmul + trenmul + carmul + ovnimul + motomul + espmul + rayomul;
43

```

Por otra parte y una vez realizado en matlab teniendo todo en orden y con todas las figuras bien representadas en el primer lenguaje, llega el momento en el cual se debe poner a prueba el código, haciendo toda la representación de lo que se trató pero ahora en C++. Viendolo desde esa perspectiva, ya no es algo tan tedioso y difícil de lograr, puesto que al tener como base lo realizado en matlab, simplemente es cuestión de ir sustituyendo valores y de ir agregando todo en orden al nuevo script. Una vez realizando todo ello, nos lleva al momento de representarlo en pantalla de salida, lo que nos da como resultado la siguiente imagen:

```
C:\Users\familia\Desktop\proyectoParadigmasFinal\proy...
1 5 7 3 -1 -1 1 3 1 1 5 7 7 5 3 3 5 3 3 5 7 0
0000110000
0001110000
0011100000
0111000000
1110000000
1110000000
0111000000
0011110000
0001110000
0011100000
0111000000
0110000000
-1-1-1-111-1-1-1-1
-1-1-1111-1-1-1-1
-1-1111-1-1-1-1-1
-1111-1-1-1-1-1-1
111-1-1-1-1-1-1-1
111-1-1-1-1-1-1-1
-1111-1-1-1-1-1-1
-1-1111-1-1-1-1-1
-1-1-1111-1-1-1-1
-1-1111-1-1-1-1-1
-1111-1-1-1-1-1-1
-111-1-1-1-1-1-1-1
energy of initial configuration : -15400
0 0 0 0 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 1 1 1 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0
0 0 1 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 1 1 1 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0
Number of iterations 1
energy of end configuration : -15400
-----
Process exited after 5.602 seconds with return value 0
Presione una tecla para continuar . . .
```

La red neuronal es capaz de reconocer las figuras plasmadas en las matrices en ambos lenguajes, llevandonos así mismo a probar la importancia del uso de Hopfield Las redes de Hopfield, ya que se usa como sistema de memoria asociativa con unidades binarias. Están diseñadas para converger a un mínimo local, pero la convergencia a uno de los patrones almacenados no está garantizada. Pero en este proyecyo nos dedicamos a hacer que funcionara tal cual era el objetivo principal, logrando así que fuese capaz de reconocer siete patrones distintos en una magnitud de 180 neuronas.

Una ventaja significativa es que las redes de Hopfield son bastante tolerantes al ruido, cuando funcionan como memorias asociativas y asi mismo también Prácticamente no existe tiempo de entrenamiento, ya que este no es un proceso adaptativo, sino simplemente el cálculo de una matriz (T).

0.3 Conclusión

Este modelo puede utilizarse como heurística para resolver problemas de optimización combinatoria, tal como fue el caso de este proyecto, su objetivo fue recibir un entrenamiento de reconocimiento de patrones con figuras, que la red fuese capaz de analizar cada una de las matrices planteadas y a su vez fuese capaz de determinar cual de todas ellas era la opción acertada. Para problemas de optimización se ha aplicado para la resolución de manipulación de grafos, por ejemplo el problema del viajante vendedor; resolución de ecuaciones, procesamiento de señales (convertidores analógico-digitales) y de imágenes, etc. Fue un gran reto trabajar con un modelo así y al mismo tiempo nos brindó mucha más experiencia tanto analítica como teórica y práctica, agregando también que nos ha dejado una grata experiencia, trabajando sobre dos lenguajes distintos con sintaxis distintas que nos enriquecieron con los distintos modos de resolver problemas de este nivel.