

Uso de redes de ‘Hopfield’ para clasificación de datos.

Alejandro F. Autor.

Resumen – Se puede reconocer de forma rápida caracteres conocidos, sin la necesidad de tener un dataset muy amplio, permitiendo que un usuario escriba una cantidad limitada de veces un carácter, y se caracterice correctamente el mismo carácter.

Hopfield – AI - Reconocimiento

I. INTRODUCCION

Para poder tener un tipo de reconocimiento, es importante la selección de interpretador correcta. Mientras se puede tener una red neuronal, ellas presentan una necesidad muy grande de datos iniciales para poder entrenarse. Gracias a esto, no es factible crear redes neuronales personalizadas para una persona, ya que se tendrían que pedir todos sus datos. Sin embargo, existe la red de reconocimiento de Hopfield. Esta es una red que se entrena a base de patrones, con la capacidad de recordarlos de forma correcta. En este papel, se va a intentar clasificar números por medio de reconocimiento de redes de Hopfield.

II. RECONOCIMIENTO

A. Bases matemáticas

El reconocimiento de datos para una red de hopfield requiere 3 pasos:

1. Preprocesamiento
2. Entrenamiento
3. Reconocimiento

Para poder tener un buen reconocimiento, se debe de tener en cuenta que los datos tienen que ser grandes y variados. Las redes de hopfield son limitadas en ese sentido, sin embargo, cuando se tienen esos elementos, sirven de forma efectiva.

El proceso de entrenamiento o creacion de matriz pqara recordar se tiene que hacer de la siguiente forma:

Documento recibido el 6 de septiembre del 2024. Este trabajo fue apoyado en parte por el Instituto Tecnológico de Estudios Superiores de Monterrey CSF. los títulos del Documento deben ser escritos en letras mayúsculas y minúsculas, no todas las mayúsculas.

$$W = \frac{\sum_{i=1}^n A_i \cdot A_i^T}{n}$$

Eq. 1. Obtención de los pesos.

Una vez obtenidos los pesos, se debe de realizar el siguiente paso para obtener un resultado:

$$R = B \cdot W$$

Eq. 2. Reconocimiento de patrones.

Donde:

W = Los pesos obtenidos

A = Conjunto de vectores de 1D para entrenar

n = Longitud de la lista para probar

B = El vector a reconocer

R = El vector resultante. Debe de resultar $R \in A$

B. Preprocesamiento

Para poder tener un buen dataset, se debe de considerar que el dataset tenga valores de -1 a 1, esto es para que se pueda tener un reconocimiento correcto. Se no ser así, el proceso de entrenamiento y recuerdo no va a ser factible.

El preprocesamiento debe de también verificar que todos los datos sean del mismo tamaño, en un arreglo de 1D. Esto significa que para una imagen, se debe de convertir a un valor de tamaño $w \times h$, donde:

- w = Grosor de la imagen
- h = Altura de la imagen

El paso final del preprocesamiento es asegurarse que la información se pueda recuperar. Por ejemplo, si se tiene un arreglo de -1 a 1, poder gratificarlo como muestra la figura 1:



Fig. 1. Los patrones de entrenamiento reordenados.

Para obtener estos datos, se esta tomando la base datos del MINST, y promediando. Después de promediarlo se toma el

resultado y se mete a una función de activación:

$$-1 \text{ si } t < 0.5$$

$$1 \text{ si } t \geq 0.5$$

Eq. 3. función de activación.

Una vez obtenidos los datos, se pueden visualizar como en la Fig. 1.

C. Entrenamiento.

Para poder entrenar el modelo, se genera un arreglo encontrado anteriormente, y se introduce a la Eq. 2. Esto permite entrenar de forma muy eficiente a la base de datos.

Una vez entrenado, se obtiene el modelo W. O bien 'weights', una matriz.

D. Llamada

Una vez entrenado el modelo, se tiene que correr la formula Eq. 2. una cantidad definida de veces, esto para poder encontrar el bajo local. Idealmente este debe de ser una cantidad baja, sin embargo puede no serlo.

En este caso, se están contemplando 5 llamadas a la misma formula, donde R se convierte en B al finalizar cada iteración.

Los resultados obtenidos son los que siguen:

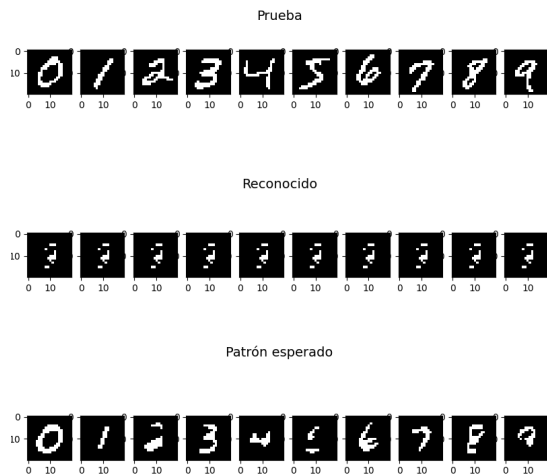


Fig. 2. Resultados de todas la interpretaciones numéricas.

E. Interpretación de datos

Como se puede observar en la Fig. 2., el modelo es un fracaso. Esto puede ser debido a varias circunstancias, pero la mas

probable es que se deba a que existe un sobre-entrenamiento de patrones.

El sobre-entrenamiento ocurre cuando gracias a la similitud de varios patrones se obtienen patrones mínimos locales. Es decir, se obtienen patrones que no deben de encontrarse en esos puntos.

Para probar este dato, se puede probar un patrón de entrenamiento menor.

F. Patrones de entrenamiento menores

Probando con solo 2 patrones, se obtienen los siguientes resultados:

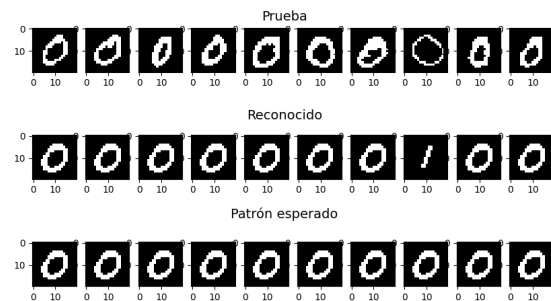


Fig. 3. Patrones de prueba con 0

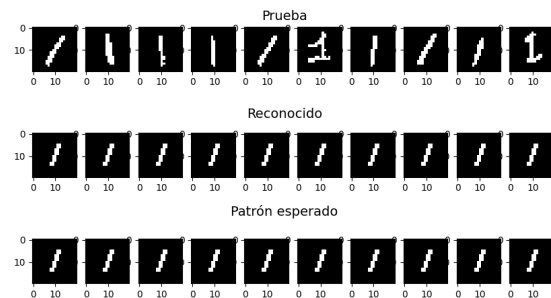


Fig. 4. Patrones de prueba con 1

Para este modelo, se obtiene un patrón de prueba

G. Resultados

Como se puede observar, los resultados con solo 2 clasificaciones son prometedoras. Al obtener un promedio, la gran mayoría de datos es contemplada, sin embargo el entrenamiento sigue siendo rápido y personalizado.

Al finalizar, se obtuvo un reconocimiento mayor al 97% correcto. Sin embargo esto solo al comparar 2 patrones.

Mientras es un muy buen numero, este método aplicado de esta forma no es posible para reconocimiento de todos los dígitos.

III. SIGUIENTES PASOS

A. *Posibles soluciones a experimentar*

Dado las limitaciones del tamaño del dataset, solo se pueden tener patrones binarios. Sin embargo, se puede mezclar números para crear una búsqueda de árbol.

La búsqueda de árbol puede mezclar distintas características de cada numero (como 7, 9 y 3) para agruparse con patrones diferentes. De esta forma, se puede crear una búsqueda y seguir teniendo aprendizaje en base de pocos ejemplos.

Otra solución es el intento de crear una imagen mas grande. Usando filtros y la información proporcionada por medio de que tan gris es una esquina del MINST, se puede crear una imagen mas grande para encontrar una mayor cantidad de características. Así los patrones serán mucho mas diferentes.

La ultima solución, es investigar el reconocimiento con valores que no sean binarios. Es decir, que se puedan introducir decimales (no solo -1 y 1). De esta forma, el modelo puede tener una precisión mayor con los datos.

IV. CONCLUSIÓN

Usando lo aprendido, se puede ver que existen redes que pueden reconocer patrones de formas muy interesantes. Sin embargo, cuentan con limitaciones. El ver y usar diferentes modelos nos sirve para conocer mejor que es lo que se tiene y lo que no. Sus casos de uso, y son herramientas que cualquier desarrollador de inteligencia artificial debe de aprender.

RECONOCIMIENTO

gracias al Prof. Gualberto por enseñarnos este modelo y sus distintas aplicaciones. Al finalizar este paper, se pueden apreciar nuevos métodos de realizar los procedimientos.