



INTELIGENCIA ARTIFICIAL

Trabajo Práctico No. 3

Redes Neuronales Artificiales

Modelo de Hopfield

Alumno: Alcover, N. Ariel.

Legajo: VINF012465

Titular Experto: Pablo Alejandro Virgolini.

Titular Disciplinar: Maria Paula Gonzalez.

Tutoría: CUATRIMESTRALES 2/24

Índice

Cuestionario.....	3
1. Hacer un análisis de los modelos neuronales artificiales, poniendo el foco en las formas de aprendizaje y preparar un breve resumen de los mismos destacando sus aplicaciones.....	3
2. Orientar el análisis anterior hacia la identificación de imágenes del tipo de las del problema abordado: modelos apropiados, ventajas y limitaciones.....	5
3. Desarrollar e implementar un prototipo destinado a la identificación de imágenes a través del modelo de Hopfield en el contexto de un caso sencillo. Describir sus características, ventajas y limitaciones frente al problema tratado y presentar un ejemplo. También resumir las dificultades encontradas.....	6
4. Analizar las coincidencias y diferencias entre el problema de fondo (línea de montaje) y las posibilidades que ofrecen el método aplicado y su prototipo. Justificar	9
Bibliografía	10

Cuestionario

1. Hacer un análisis de los modelos neuronales artificiales, poniendo el foco en las formas de aprendizaje y preparar un breve resumen de los mismos destacando sus aplicaciones.

Los modelos neuronales artificiales son sistemas computacionales inspirados en el funcionamiento del cerebro humano, diseñados para realizar tareas específicas como reconocimiento de patrones, toma de decisiones y predicciones. Los métodos de aprendizaje son fundamentales para entrenar estos modelos.

Las redes neuronales artificiales se componen de varias capas que trabajan en conjunto para realizar tareas de aprendizaje. Cada tipo de capa tiene un papel específico en el proceso de aprendizaje y en la transformación de los datos a través de la red. Las más comunes son: de Entrada, Ocultas y de Salida.

Basada en las lecturas del CANVAS, me voy a centrar en la siguiente clasificación para poder desarrollar los diferentes enfoques en el aprendizaje de modelos neuronales artificiales.

Aprendizaje Supervisado: Este método utiliza un conjunto de datos etiquetado, donde cada entrada tiene una salida correspondiente conocida. El objetivo es que la red neuronal aprenda a mapear correctamente las entradas a sus salidas esperadas ajustando los pesos a través de un proceso de retro propagación. Las redes neuronales ajustan sus pesos para minimizar el error en la predicción de salidas conocidas a partir de entradas dadas.

- a. Aplicaciones: Reconocimiento de voz, clasificación de imágenes, análisis de sentimientos, detección de fraudes, entre otros.
- b. Ventajas: Alta precisión en tareas bien definidas con datos etiquetados.
- c. Limitaciones: Requiere una gran cantidad de datos etiquetados, lo que puede ser costoso y laborioso de obtener.

Aprendizaje No Supervisado: No se proporcionan etiquetas para las entradas. La red busca patrones o estructuras en los datos, como agrupamientos o clustering, a partir de la

información no etiquetada, con el fin de identificar relaciones ocultas o características en los datos. Este método permite que la red aprenda sin etiquetas, buscando patrones en los datos de entrada.

- a. Aplicaciones: Agrupamiento de clientes en marketing, detección de anomalías, reducción de dimensionalidad (e.g., PCA), etc.
- b. Ventajas: No necesita datos etiquetados, lo que lo hace útil en situaciones donde la etiquetación es difícil o costosa.
- c. Limitaciones: Puede ser difícil interpretar los resultados y la calidad del aprendizaje depende de la naturaleza de los datos.

Aprendizaje Híbrido: Este enfoque combina elementos del aprendizaje supervisado y no supervisado en una sola red. Las diferentes capas pueden utilizar diferentes estrategias de aprendizaje, lo que permite aprovechar las ventajas de ambos métodos para mejorar el rendimiento de la red.

- a. Aplicaciones: Procesamiento de lenguaje natural, sistemas de recomendación, etc.
- b. Ventajas: Permite aprovechar lo mejor de ambos mundos, adaptándose a problemas complejos que pueden beneficiarse de ambos enfoques.
- c. Limitaciones: La complejidad del diseño y entrenamiento puede ser mayor, y el acceso a datos de calidad sigue siendo crucial.

Aprendizaje por Refuerzo: Este método se basa en la interacción entre el agente (la red) y su entorno. En lugar de aprender de un conjunto de datos etiquetado, el modelo recibe retroalimentación en términos de recompensas o penalizaciones a partir de sus acciones. El objetivo es maximizar la recompensa a largo plazo, lo que lo convierte en un enfoque muy útil en situaciones donde la secuencia de decisiones importa.

- a. Aplicaciones: Juegos (e.g., AlphaGo), robótica, navegación autónoma, optimización de sistemas (como gestión de inventarios).
- b. Ventajas: Capacidad para aprender estrategias de largo plazo y adaptarse a entornos dinámicos.

- c. Limitaciones: Necesita una cantidad significativa de entrenamiento y puede ser ineficiente en entornos donde las recompensas son escasas.

- 2. Orientar el análisis anterior hacia la identificación de imágenes del tipo de las del problema abordado: modelos apropiados, ventajas y limitaciones.

Cada uno de los enfoques tiene sus propias aplicaciones y es adecuado para diferentes tipos de problemas.

Cuando se trata de la identificación de imágenes, el aprendizaje supervisado es el enfoque más común debido a la disponibilidad de grandes conjuntos de datos etiquetados, como ImageNet para tareas de clasificación de imágenes. Las redes neuronales convolucionales (CNN) son típicamente las mejores para este tipo de tareas.

- a. Modelos Apropriados:

→ Redes Neuronales Convolucionales (CNN): Especializadas en el procesamiento de datos que tienen una estructura de cuadrícula, como las imágenes. Ejemplos incluyen arquitecturas como ResNet y VGG.

→ Transfer Learning: Utilizar modelos pre-entrenados en grandes datasets para adaptarlos a tareas específicas de identificación de imágenes, lo que reduce el tiempo y el costo del entrenamiento.

- b. Ventajas:

Robustez: Las CNN son particularmente efectivas para extraer características jerárquicas de las imágenes, lo que mejora la precisión del modelo.

Generalización: Los modelos pre-entrenados pueden adaptarse a nuevos dominios con un esfuerzo razonable en el re-etiquetado.

- c. Limitaciones:

Requerimientos de Datos: Aunque el uso de Transfer Learning puede mitigar este problema, en general se necesita una cantidad significativa de imágenes etiquetadas para un buen desempeño.

Computación Intensiva: Tanto el entrenamiento como la inferencia pueden ser computacionalmente costosos, requiriendo hardware avanzado.

La elección del modelo de aprendizaje correcto depende del problema específico y del tipo de datos disponibles. Para la identificación de imágenes, el aprendizaje supervisado con redes neuronales convolucionales demuestra ser uno de los métodos más efectivos, aunque la recolección de datos y el costo computacional son desafíos a considerar. Por otro lado, el aprendizaje no supervisado y reforzado ofrece oportunidades monetarias en áreas donde los datos etiquetados son limitados.

3. Desarrollar e implementar un prototipo destinado a la identificación de imágenes a través del modelo de Hopfield en el contexto de un caso sencillo. Describir sus características, ventajas y limitaciones frente al problema tratado y presentar un ejemplo. También resumir las dificultades encontradas.

Afronté el desafío pensando en único vector de 100 caracteres. Armé el vector correspondiente al patrón deseado, lo multiplico por su traspuesto y obtengo la matriz de pesos. Seguidamente, genero un vector aleatorio como entrada a la red neuronal y en no más de dos pasadas, generalmente una, ya obtengo el patrón deseado. Este desarrollo queda codificado ahora en la clase *HopfieldOptimo*.

Imagen del modelo

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	1	0	0	0
2	0	0	0	1	1	1	1	0	0	0
3	0	0	1	0	0	0	0	1	0	0
4	0	1	1	0	0	0	0	1	1	0
5	0	1	1	0	0	0	0	1	1	0
6	0	0	1	0	0	0	0	1	0	0
7	0	0	0	1	1	1	1	0	0	0
8	0	0	0	0	1	1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

En la clase virtual, para la implementación, desean que el modelo de aprendizaje conste de más de un patrón. Lo cual estaría de acuerdo con la utilidad del modelo, a la vez que su limitación; ya que Hopfield es útil para reconocer varios patrones, pero no infinitos... además, los patrones deben cumplir algunas condiciones, como ser ortogonales o variar en, al menos $N/2$, con N número de neuronas. Se puede observar en la figura que se trata de dos círculos idénticos al primero, corridos del centro.

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	1	0	0	0	0	0
1	0	0	1	1	1	1	0	0	0	0
2	0	1	0	0	0	0	1	0	0	0
3	1	1	0	0	0	0	1	1	0	0
4	1	1	0	0	0	0	1	1	0	0
5	0	1	0	0	0	0	1	0	0	0
6	0	0	1	1	1	1	0	0	0	0
7	0	0	0	1	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	0	0	0
3	0	0	0	0	1	1	1	1	0	0
4	0	0	0	1	0	0	0	0	1	0
5	0	0	1	1	0	0	0	0	1	1
6	0	0	1	1	0	0	0	0	1	1
7	0	0	0	1	0	0	0	0	1	0
8	0	0	0	0	1	1	1	1	0	0
9	0	0	0	0	0	1	1	0	0	0

Al sumar estos nuevos patrones, el modelo desarrollado según la clase *Hopfield3Ciclos*, no arrojaba ningunos de los patrones circulares.

Imagen 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]

```

Le di vueltas al código, aunque no encontraba que tenga un error. Con el espíritu renovado, inicié nuevamente la codificación de los tres círculos, ahora con vectores de 10 caracteres, de esta manera podría seguir la matemática más cómodamente, pensando que el error podría estar en la creación de la matriz de pesos, un monstruo de 100 x 100. Esta codificación es más laboriosa ya que debo generar 10 matrices de peso individuales, para luego combinar los resultados y hacer más simple su comparación. su avance se puede ver en la clase *HopfieldIndividual*.

Mi sorpresa se dio cuando nuevamente, en la línea 2, empecé a ver el mismo resultado. En lugar de tener 4 unos, aparecían 2 unos. Analizando los vectores, cambio mi hipótesis, infiriendo que observo que los resultados obtenidos ubican los "1" de acuerdo con las posiciones donde más se repiten

```

int[] patron4a = {-1, -1, 1, 1, -1, -1, 1, 1, -1, -1};
int[] patron4b = {-1, 1, 1, -1, -1, 1, 1, -1, -1, -1};
int[] patron4c = {-1, -1, -1, 1, 1, 1, 1, 1, 1, -1};
// espero 1 en posiciones 3, 4, 6, 7 y 8

```

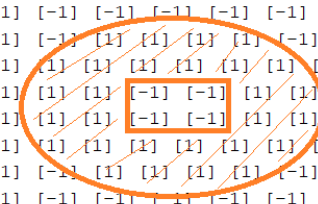
Confirmada la hipótesis, ya que estoy reproduciendo la imagen 1 vector por vector (o línea por línea). Cambio el dibujo del círculo, haciendo trazos más gruesos, con la esperanza de mejorar la repetición de "1" (unos) en cada vector.

Imagen 2


```

Elija opción
2
No hubo cambios
coincide con el patrón
[-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]
Elija opción

```



Modifico los vectores de la clase *Hopfield3Ciclos* para ver que resultados obtengo. Efectivamente, en no más de 2 repeticiones, el modelo se estabiliza y siempre encuentra el círculo óptimo.

4. Analizar las coincidencias y diferencias entre el problema de fondo (línea de montaje) y las posibilidades que ofrecen el método aplicado y su prototipo. Justificar.

En líneas generales, no es una solución adecuada. El modelo de Hopfield se utiliza con otros fines, como desarrollamos en el FORO. Otro ejemplo podría ser para digitalizar texto a partir de una imagen, como puede ser al escanear un texto.

Si bien desarrollé un código que siempre encuentra el orificio del motor central, no es una solución elegante, según lo veo, estoy forzando el modelo para llegar al resultado deseado. Como conversamos por chat.

“desean que el modelo de aprendizaje conste de más de un patrón.. lo cual estaría de acuerdo con la utilidad del modelo, a la vez que su limitación; ya que Hopfield es útil para reconocer varios patrones, pero no infinitos... además, deben ser ortogonales!

yo interpreto que esto sería a costa de la optimización del ejercicio, ya que podría hacer varias pasadas y arribar a un patrón no deseado, es decir uno de los círculos (llamémoslo) señuelo!”

Finalmente, al optimizar el círculo para dejar tres vectores en donde la mayor cantidad de puntos siempre se encuentran en el círculo central, estoy forzando el modelo a dar la respuesta que quiero.

<https://github.com/arielna/Inteligencia-Artificial-2024.git>

Bibliografía

<https://www.ibm.com/es-es/topics>

Inteligencia Artificial, Un Enfoque Moderno. S. Russell y P. Norvig. Segunda Ed. Pearson, España (2004).