

Memoria asociativa

IMPLEMENTACIÓN DE UNA MEMORIA ASOCIATIVA POR
MEDIO DE UNA RED NEURONAL ARTIFICIAL

Planteamiento

Se quería diseñar un programa para ilustrar el potencial de las redes neuronales en el reconocimiento y aprendizaje de patrones. Específicamente, su capacidad para inferir relaciones entre patrones para recordar y asociarlos con otros.

En este caso, ingresar un conjunto de patrones similares a letras y ver si los estímulos retornaban la letra más similar.

Base conceptual

La memoria asociativa es el almacenamiento y recuperación de información por asociación con otras informaciones.

Un dispositivo de almacenamiento de información se llama memoria asociativa si permite recuperar información a partir de conocimiento parcial de su contenido, sin saber su localización de almacenamiento. A veces también se le llama memoria de direccionamiento por contenido

Los computadores tradicionales no usan este direccionamiento; se basan en el conocimiento exacto de la dirección de memoria en la que se encuentra la información.

Sin embargo, se cree que el cerebro humano no actúa así. Si queremos recordar el nombre de una persona, no nos sirve saber que fue el nombre número 3274 que aprendimos. Es más útil saber que su nombre empieza y termina por 'N' y que es un famoso científico inglés. Con esta información, es casi seguro que recordaremos exitosamente a "Newton".

Las memorias asociativas son una de las redes neuronales artificiales más importantes con un amplio rango de aplicaciones en áreas tales como: Memorias de acceso por contenido, identificación de patrones y control inteligente.

Una memoria asociativa puede almacenar información y recuperarla cuando sea necesario, es decir, una red retroalimentada, cuya salida se utiliza repetidamente como una nueva entrada hasta que el proceso converge. Puede recuperar dicha información basándose en el conocimiento de parte de ésta (clave). El patrón clave puede ser una versión con ruido de un patrón memorizado, es decir, que difiere de él en pocas componentes. La memoria humana recuerda a una persona aunque vaya vestida de forma diferente o lleve gafas.

Desarrollo de la aplicación

Se comenzó investigando la base conceptual y tras haber formado la interfaz se procedió a implementar una estructura de clases similar a lo que sería un grafo conexo con arcos no dirigidos. Esto es, que cada nodo (neurona de tipo perceptrón) está conectado a todos los otros nodos por solo un arco (peso).

Siendo que la especificación de la aplicación demandaba patrones de 4x4, la red resultante debía tener 16 neuronas y 120 sinapsis.

A manera general: Los patrones se van introduciendo en el panel respectivo y se guardan en una lista de Sites (una clase que hereda de la clase Polygon y que contiene además el valor del site (encendido: 1, apagado: -1)). Al terminar el patrón puede ser agregado a la memoria, lo que activa el aprendizaje de la red por medio del método `learnpattern()`.

Tras guardar los patrones, puede comprobarse si la memoria lo ha aprendido satisfactoriamente al introducir un estímulo en el panel de estímulo y comprobar el resultado.

El método `learnpattern()` presenta el valor del site a su neurona correspondiente (teniendo en cuenta que cada site esta numerado y se relaciona con una neurona con ese mismo número) y comienza a iterar:

1. Mientras halla meta error (error en al menos una parte de la red) o no se alcance el máximo de meta iteraciones:
 - 1.1. Por cada neurona de la red:
 - 1.1.1. Mientras halla error (diferencia entre el valor obtenido y el deseado) o no se alcance el máximo de iteraciones:
 - 1.1.1.1. Por cada peso que entra a la neurona actual:
 - 1.1.1.1.1. Se actualizan el peso de la entrada al sumarle la multiplicación de la taza de aprendizaje, por la resta de 1, menos la multiplicación de la salida deseada por la salida actual, por la multiplicación de la deseada con el valor de la entrada correspondiente. Siendo que esta es la ecuación de actualización para perceptrones con salidas bipolares.

El final del ciclo mientras ocurre si en ninguno de los puntos ha habido error o se halla iterado demasiadas veces. La autocorrección que el perceptron aplica a si mismo ahora influye en todos los otros perceptrones, de forma que cada perceptron está relacionado con los otros y su respuesta dependerá de las respuestas de todos.

Nota: ver el código del programa para mejor referencia.

Uso de la interfaz

Botones:

Limpiar:

Los botones limpiar permiten reiniciar los paneles de entrenamiento y estímulo, dejándolos en blanco.

Reset:

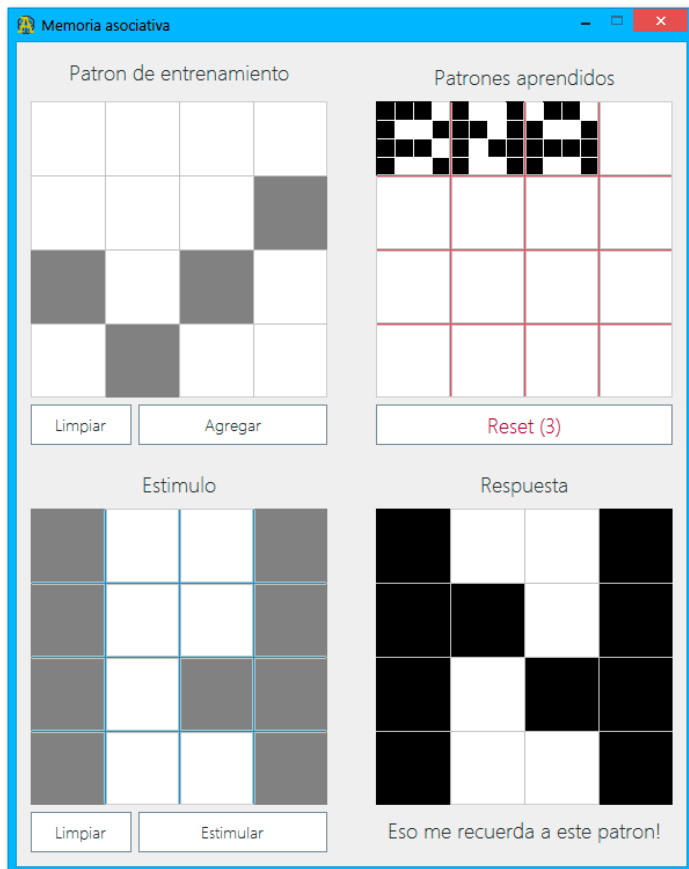
Fuerza a la memoria a olvidar todos los patrones aprendidos y reiniciar sus pesos.

Agregar:

Suministra el patrón a la red y ejecuta su aprendizaje. El tiempo que la red tarda en aprender un patrón es variable y puede ser instantáneo o demorar varios segundos.

Estimular:

Presenta el patrón de estímulo y pregunta a cada neurona por su respuesta. El patrón reconstruido se muestra en el panel de respuesta.



Paneles:

El panel estímulo y de entrenamiento responden al ratón. Click izquierdo para marcar una casilla, click derecho para desmarcarla.

El panel de aprendidos muestra una miniatura de todos los patrones aprendidos hasta el momento.

El panel de respuesta es la reconstrucción que la memoria ha hecho a partir del patrón de estímulo.