Resumen del Capítulo 2 de "Introduction To The Theory of Neural Computation" de Hertz, Krogh y Palmer (1991).

Rodrigo García Núñez

Universidad Autónoma Metropolitana

Febrero 2025

1 Capacidad de Almacenamiento

Retomando el capitulo 2 del texto y continuando con la sección de capacidad de almacenamiento.
Tomando en cuenta la siguiente expresión

$$C_i^v \equiv -\epsilon_i^v (1/N) \sum_j \sum_{\mu \neq v} \epsilon_i^m u \epsilon_j^\mu \epsilon_j^v \tag{1}$$

Esto solo son $-\epsilon_i^v$ veces el término de interferencia en la regla de estabilidad. Si C_i^v es negativo, entonces el término de interferencia tiene el mismo signo que el patrón deseado ϵ_i^v, por lo que no afecta. Por otro lado, si C_i^v sí es positivo, y mayor que 1, entonces sí cambiará el signo de h_i^v y vuelve al patrón de la unidad i inestable. Si se inicia la red en estado de memoria deseado ϵ_i^v, no se quedará ahí.

Ahora, es momento de considerar patrones aleatorios, con igual probabilidad de que ϵ_i^v sea +1 o -1. Con esto, podemos estimar la probabilidad de que un patrón arbitrario sea inestable. A esta probabilidad la llamamos P_{error} y la representamos de la siguiente forma:

$$P_{error} = Prob(C_i^v > 1) \tag{2}$$

Nótese que P_{error} aumenta cuando incrementamos la cantidad de p patrones que queremos almacenar en la red. Para encontrar la máxima cantidad de patrones que podemos almacenar en una red, debemos adoptar un criterio de aceptación de rendimiento, por ejemplo $P_{error} < 0.01$.

Cómo se puede apreciar, P_{error} depende de la cantidad de unidades N y p,

el número de patrones que queremos almacenar. Ahora C_i^v es $1/N$ veces la suma de Np números aleatorios, en donde es igual de posible que sean +1 o -1, como una distribución binomial con promedio 0 y una varianza $\sigma = p/N$. Sin embargo, asumimos que p y N son grandes, esto podemos aproximarlo como una distribución Gaussiana, con promedio y varianza iguales, como se muestra en la figura 1.

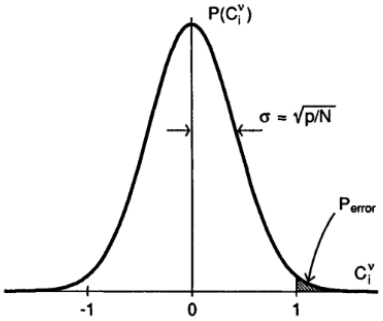

Figure 1: Distribución de los valores del término de interferencia C_i^v. Para p patrones aleatorios y N unidades, se obtiene una distribución Gaussiana con varianza $\sigma = p/N$. Lo sombreado es la probabilidad de error por bit.

Nótese que la probabilidad P_{error} en la que C_i^v sobrepasa el valor de 1 se encuentra sombreada en la figura 1. Dicho esto, para calcular P_{error} utilizamos la siguiente fórmula

$$P_{error} = 1/\sqrt{2\pi\sigma}\int_1^\infty e^{-x/2\sigma}dx = 1/2[1-erf(1/\sqrt{2\sigma})] = 1/2[1-erf(\sqrt{N/2p})] \tag{3}$$

En donde la **Función de error** está definida por

$$erf(x) \equiv 2/\sqrt{\pi}\int_0^x exp(-u)du \tag{4}$$

La siguiente tabla muestra los valores de p/N para varios valores de P_{error}. Por ejemplo, si tomamos el criterio de que $P_{error} < 0.01$, entonces tendríamos que $p_{max} = 0.15N$. Este cálculo nos dice la estabilidad inicial del modelo. Por ejemplo, si tomamos que $p < 0.185N$, entonces aproximadamente el 1% de los bits iniciales serán inestables. Esto, aunque parece inofensivo, podría tener el efecto de que si iniciamos desde un patrón ϵ_i^v y el 1% de los bits se invierten, entonces podría pasar que algunos otros bits más se inviertan también. En el peor de los casos, puede ocurrir un fenómeno de avalancha en el que cada vez más bits se invierten y el sistema termine en un estado desconocido. Dicho esto, se puede notar que los valores presentes en la tabla 1 son valores elevados, por lo

P_{error}	p_{max}/N
0.001	0.105
0.0036	0.138
0.01	0.185
0.05	0.37
0.1	0.61

Table 1: Valores de P_{error} con su respectivo valor de p_{max}/N

que se necesitan valores más pequeños para mantener una estabilidad aceptable en el modelo.

Una definición alternativa de la capacidad de almacenamiento insiste en que la mayoría de los patrones sean recuperados perfectamente. Dado que cada patrón consta de N bits, necesitamos que $P_{error} < 0.01/N$ para que los N bits sean recuperados correctamente con un 99% de probabilidad. Esto implica que $p/N \to 0$ mientras que $N \to \infty$, así que se usa la expansión asintótica de la función de error, tal que

$$1 - erf(x) \to e^{-x}/\sqrt{\pi x} \text{ (cuando } x \to \infty) \tag{5}$$

para obtener que

$$log(P_{error}) \approx -log2 - N/2P - (1/2)log\pi - (1/2)log(N/2p) \tag{6}$$

De modo que la condición $P_{error} < 0.01/N$ se convierte en

$$-log2 - N/2P - (1/2)log\pi - (1/2)log(N/2p) < log0.01 - logN \tag{7}$$

pero, si solo nos concentramos en los términos principales para un N grande

$$N/2P > logN \tag{8}$$

Tal que el mayor número de patrones que se pueden almacenar en este caso es $p_{max} = N/2logN$. Si fuéramos más estrictos y decidiéramos que queremos que todos los patrones sean perfectamente recuperables, entonces necesitaríamos recuperar Np bits correctamente con, al menos, 99% de probabilidad, y también que $P_{error} < 0.01/pN$. Entonces la expresión 8 cambia a

$$N/2P > log(Np) \tag{9}$$

lo que indica que $p_{max} = N/4logN$ debido a que $log(Np) \sim logN^2 = 2logN$.

Nótese que se asume que para los casos de perfecta recuperación que los C_i^v son independientes de un orden.

En resumen, la capacidad de almacenamiento p_{max} es proporcional a N, más nunca es mayor a 0.138N, si es que estamos dispuestos a aceptar un pequeño porcentaje de error en cada patrón, pero si insistimos en que la mayoría de los patrones sean perfectamente recuperables, entonces p_{max} es proporcional a $N/logN$.

Sin embargo, en la vida real los patrones no son generalmente aleatorios. El modelo Hopfield se estudia con patrones aleatorios por conveniencia matemática; sin embargo, también hay casos en los que los patrones están correlacionados. Pero, para el caso en el que los patrones son ortogonales (sus productos escalares son iguales a 0), entonces no se tiene ningún tipo de interferencia, tal que $C_i^v = 0$ para todo patrón v y unidad i.

$$\sum_j \epsilon_j^\mu \epsilon_j^v = 0 \text{ para todo } \mu \neq v \tag{10}$$

En estos casos, la capacidad de memoria p_{max} sería igual a N porque a lo mucho se pueden construir N cadenas de bits mutuamente ortogonales de tamaño N. Sin embargo, la memoria útil tiende a ser más pequeña. Si intentamos guardar N patrones ortogonales con la regla de Hebb, todos los estados resultan estables; el sistema se queda en donde empieza, dando como resultado un modelo inútil como memoria. Esto pasa porque las condiciones de ortogonalidad tienden a

$$w_{ij} = \begin{cases} 1, & \text{si } i = j \\ 0, & \text{de otro modo} \end{cases} \tag{11}$$

tal que cada unidad está conectada solo a sí misma. Para poder obtener una medida de la capacidad p_{max}, es necesario sugerir una zona de atracción finita alrededor de cada patrón deseado. Esto nos da una capacidad útil ligeramente menor que N.

$$H = -1/2 \sum_{ij} w_{ij} S_i S_j \tag{12}$$

2 Bibliografía

References

[1] John H., Anders K. y Richard G. P. (1992). Introduction to the theory of Neural Computation (pp. 17 - 20). CRC Press.