REDES NEURONALES - PRÁCTICA 5 - 2024

Aprendizaje no supervisado

Maximiliano Gatto

Instituto Balseiro (UNCuyo - CNEA) - Bariloche, Río Negro, Argentina

maximiliano.gatto@ib.edu.ar

27 de octubre de 2024

1. Introducción

En esta práctica se analizaron diferentes técnicas de aprendizaje no supervisado. En particular, se implementó la regla de hoja y el "future mapping". Se verificó el funcionamiento de los algoritmos y se analizaron las convergencias comparando con los resultados obtenidos en las clases teóricas. Todos los ejercicios fueron implementados mediante un script en `Python`, cuyo código está disponible en este enlace.

2. Resultados

Ejercicio 1

Se considera una red de una capa lineal, con 4 entradas y 1 salida, en donde la salida de la red está dada por $V = \sum_{j=1}^{4} \omega_J X_j$, donde X_j son las entradas y ω_j son los pesos sinápticos.

La distribución de probabilidad de los valores de entrada de corresponden con una distribución gaussiana multidimensional con media 0 dada por

$$P(X) = \frac{1}{(2\pi)^2 |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\overline{X}^T \Sigma^{-1} \overline{X}\right),$$

cuya matriz de covarianza es

$$\Sigma = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}.$$

Se utilizaron valores aleatorios pequeños $[-0.01, 0.01]$ distribuidos uniformemente como condición inicial para los pesos ω, y se los modificó según la regla de oja analizada en la clase teórica

$$\Delta\omega_{i,j} = \eta V (X_i - V\omega_j),$$

donde se utilizó arbitrariamente una taza de aprendizaje $\eta = 0.005$.

Según lo estudiado en la clase teórica, siguiendo la regla de oja, los pesos obtenidos luego del aprendizaje convergen en media al autovector normalizado correspondiente al autovalor más grande de la matriz de covarianza Σ. En este caso, el autovector correspondiente al autovalor más grande de Σ es $\overline{\chi}_\Sigma = [0.5, 0.5, 0.5, 0.5]$.

Se utilizaron 10^7 ejemplos obtenidos de la distribución gaussiana multidimensional. Los resultados obtenidos para los pesos según el número de ejemplos o épocas del proceso de aprendizaje se muestran en la Figura 1.

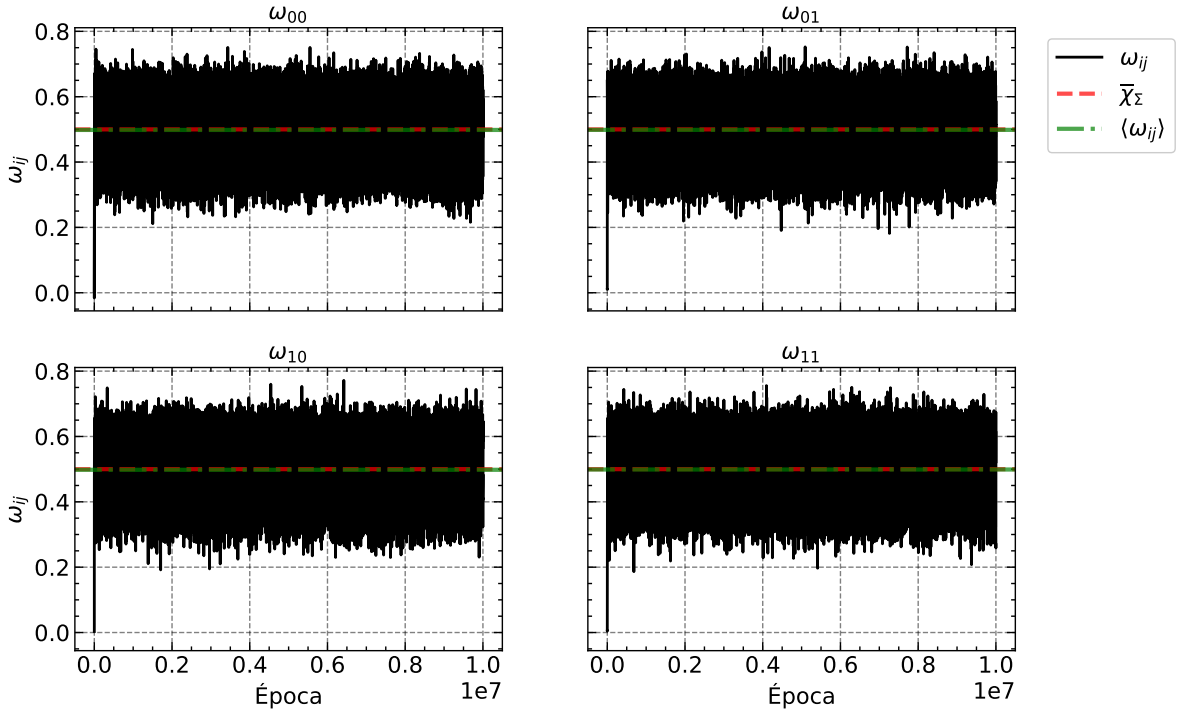

Figura 1: *evolución de los pesos sinápticos durante el proceso de aprendizaje. La línea negra representa el valor de cada una de las componentes de los pesos, la línea de trazos roja representa el valor del autovector en cada componente y la linea de trazo y punto verde representa el valor medio del peso. Se observa que el valor de los pesos es ruidoso, sin embargo el promedio de los pesos converge al autovector de mayor autovalor.*

Ejercicio 2

Se consideró una red neuronal de Kohonen para analizar la técnica de *feature mapping*, en donde en lugar de solo modificar la unidad ganadora, también se modifican los vecinos. En este ejercicio, se utilizaron 2 neuronas de entradas y 10 neuronas de salidas dispuestas sobre una línea.

Los datos de entrada de la red se extrajeron de una distribución de probabilidad dada por

$$P(\bar{\xi}) = P(r,\theta) = \begin{cases} C \text{ si } r \in [0.9, 1.1], \theta \in [0, \pi] \\ 0 \text{ sino} \end{cases}$$

donde C es una constante de proporción.

Notar que este dominio se corresponde con un "semi anillo", en donde la probabilidad es uniforme. Para obtener datos con distribución uniforme en este dominio, se generaron datos en coordenadas cartesianas (x, y) con distribución uniforme en $x \in [-1.1, 1.1]$ e $y \in [0, 1.1]$, y se descartaron aquellos puntos que no se encontraban en el semi anillo. Los valores ω_j utilizados como condición inicial se muestran en la Figura 2.

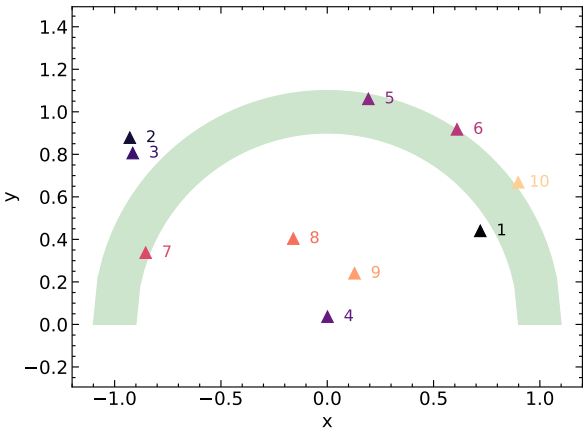

Figura 2: *valores iniciales de los pesos sinápticos de la red de Kohonen.*

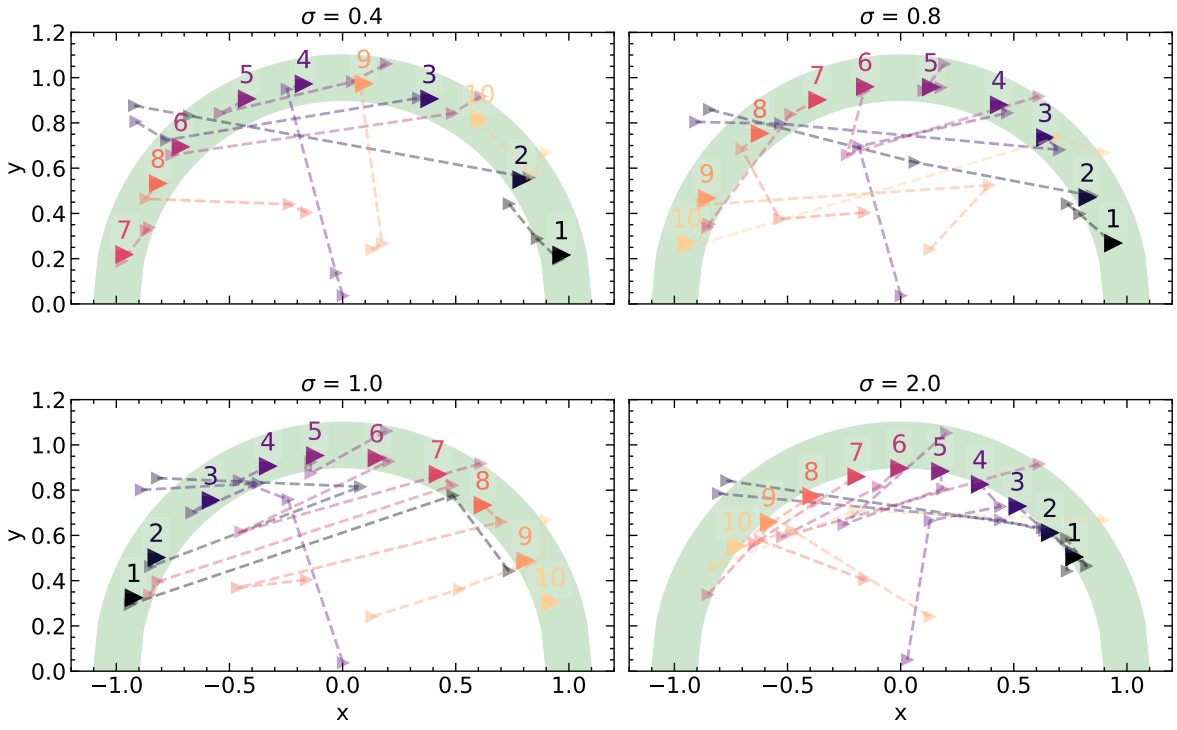

Figura 3: *evolución en función de los ejemplos o iteraciones para diferentes valores de σ. Se muestra la evolución con el número de ejemplos partiendo desde la semilla y las iteraciones correspondientes a 10^2, 10^5 y 10^7. Los colores muestran la salida analizada según la etiqueta de la figura.*

Como función de vecindad, se utilizó

$$\Lambda(i, i^*) \propto e^{-\frac{(i-i^*)^2}{2\sigma^2}}$$

donde i se corresponde i-ésima neurona (ω_i), mientras que i^* es la neurona que se encuentra más cerca del valor de entrada.

Para realizar esta red, se utilizó una taza de aprendizaje de $\eta = 0.1$ y se evaluaron diferentes valores de σ (0.4, 0.8, 1 y 2). Como la distancia entre los "primeros vecinos" es 1, un valor de $\sigma < 1$ implica una menor corrección sobre los vecinos, mientras que un valor de $\sigma \geq 1$ corrige sobre los vecinos, siendo mayor los vecinos afectados mientras mayor sea el valor de σ.

Los resultados obtenidos para 10^7 ejemplos, así como la evolución en función de los ejemplos o iteraciones se muestran en la Figura 3.

En la Figura 3 se observa que para $\sigma = 0.4$ la red no logra ordenar de manera correcta las salidas ya que los vecinos se encuentran separador por $\sim 2\sigma$, por lo que la función de vecindad tiende a 0. Para $\sigma = 0.8$ se observa que la red logra ordenar de manera correcta las salidas, sin embargo, la convergencia es más lenta que para $\sigma = 1$. Para $\sigma = 2$ se observa que la red logra ordenar de manera correcta las salidas, pero las salidas se agrupan en el centroide del dominio. Además, lo que se nota es que para $\sigma = 1$, el orden es contrario a los casos anteriores. Esto depende de como el algoritmo modifica los pesos según la cercanía según el σ dado, pero lo importante es que estén ordenados ya sea en una dirección u otra.