Redes Neuronales
Práctica 5 - Aprendizaje no supervisado

F. M. Cabrera

13 de mayo de 2021

Ejercicio 1

Tenemos una red de una capa lineal con una entrada $\vec{\xi}$ de 4 componentes y una salida lineal. La ecuación de salida viene dada por

$$V = \sum_{j=1}^{4} \omega_j \xi_j. \tag{1}$$

La distribución de probabilidad de las entradas es una distribución Gaussiana con matriz de correlación Σ

$$P\left(\vec{\xi}\right) = \frac{1}{(2\pi)^2 \sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}\vec{\xi}^t \Sigma^{-1} \vec{\xi}\right) \tag{2}$$

donde se utilizó

$$\Sigma = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}. \tag{3}$$

Los pesos ω_j de la red fueron inicializados aleatoriamente con un distribución uniforme entre $-0{,}01$ y $0{,}01$ y fueron actualizados mediante la regla de Oja

$$\Delta\omega_j = \eta V \left(\xi_j - V\omega_j\right) \tag{4}$$

donde η es el *learning rate*.

Se realizo este aprendizaje para 5000 entradas, utilizando $\eta = 0{,}001$. Los resultados obtenidos pueden observarse en la Figura 1. A la izquierda se observa la evolución de las cuatro componentes de pesos de la red. Sabemos que el aprendizaje dado por la regla de la Oja produce que el vector $\vec{\omega}$ dado por los pesos de la red, evolucione hacia la dirección del autovector de mayor autovalor de la matriz Σ. Dicha matriz posee 2 autovalores, 5 y 1. El autovalor 5 no tiene degeneración y su autovector es $\vec{v} = (0{,}5, 0{,}5, 0{,}5, 0{,}5)^t$. De esta manera, se observa que efectivamente los pesos de la red evolucionan alineándose a dicho autovector, como es esperado.

Por otro lado, otra de las caracteristicas del aprendizaje dado por la regla de Oja es que $\vec{\omega}$ tiende a un vector unitario, es decir que $|\vec{\omega}|$ tiende a 1, lo cual se observa a la derecha de la Figura 1. Por último, simplemente a fin de visualización, en la Figura 2 se observa un gráfico 3D con tres de las componentes de $\vec{\omega}$ a lo largo del aprendizaje.

Ejercicio 2

Para este ejercicio se tiene un red neuronal de Kohonen con dos neuronas de entrada. Se utilizaron dos neuronas de salidas, dispuesta sobre una linea entre los puntos $\vec{x}_1 = (-1, 0{,}5)$ y $\vec{x}_1 = (1, 0{,}5)$ de forma equiespaciada.

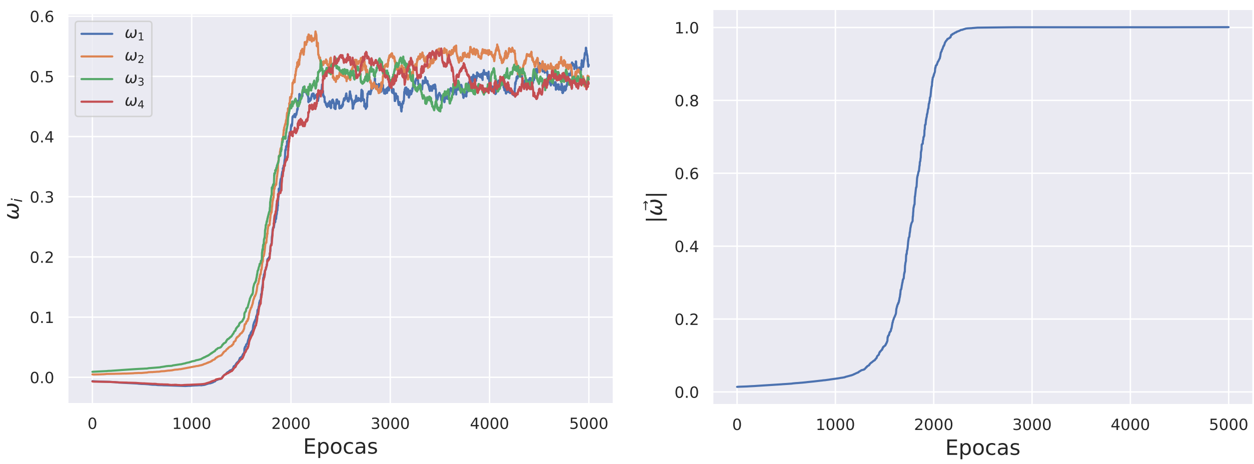

Figura 1: A la izquierda, se observa la evolución a lo largo de las épocas de las cuatro componentes ω_j de la red utilizada mientras que a la derecha se observa la evolución de la norma $|\vec{\omega}|$. Tal y como predice la teoría, el vector $\vec{\omega}$ se alinea en la dirección del autovector de mayor autovalor de la matriz Σ, que en este caso corresponde a $\vec{v} = (0{,}5, 0{,}5, 0{,}5, 0{,}5)^t$. De la misma manera, $\vec{\omega}$ evoluciona de manera que $|\vec{\omega}| = 1$.

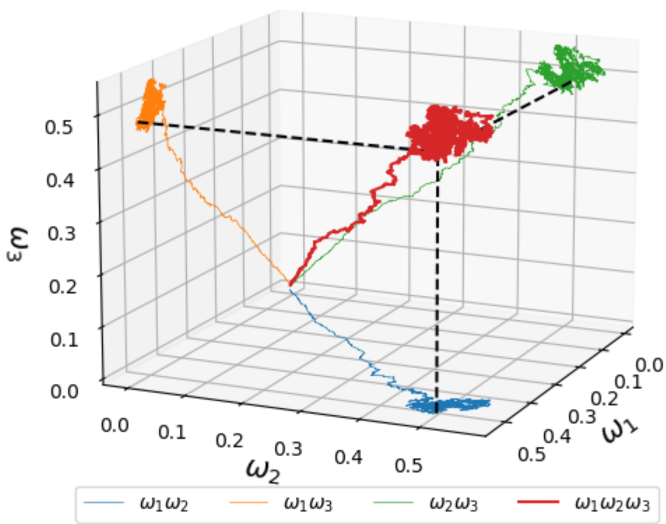

Figura 2: Evolución de los pesos ω_1, ω_2 y ω_3 durante el aprendizaje. Se observa que los mismos se alinean en la dirección del autovector de mayor autovalor de la matriz Σ, $\vec{v} = (0{,}5, 0{,}5, 0{,}5, 0{,}5)^t$.

Las entradas $\vec{\xi}$ de la red fueron generadas con una distribución

$$P\left(\vec{\xi}\right) = P\left(t, \theta\right) = \begin{cases} \text{cte} & \text{si} \qquad r \in [0{,}9, 1{,}1] \;\; \text{y} \;\; \theta \in [0, \pi] \\ 0 & \text{en otro caso} \end{cases} \tag{5}$$

es decir que, las entradas generadas estas distribuidas uniformemente en media corona de radio menor 0,9 y radio mayor 1,1.

En cada paso del aprendizaje, es necesario obtener un indice i^* ganador, el cual corresponde a la unidad de pesos mas cercano a la entrada $\vec{\xi}$. Esta condicion puede expresarse como

$$\left\|\vec{\omega}_{i^*} - \vec{\xi}\right\| \leq \left\|\vec{\omega}_i - \vec{\xi}\right\| \qquad \forall i \tag{6}$$

donde $\vec{\omega}_i$ denota a los pesos de la salida i.

Una vez obtenido el indice ganador i^*, la regla de aprendizaje para los pesos es

$$\Delta\omega_{ij} = \eta\Lambda\left(i, i^*\right)\left(\xi_j - \omega_{ij}\right) \tag{7}$$

donde η es el *learning rate* y Λ en la *función vecindad*, que para este ejercicio se utilizo una *función vecindad Gaussiana* dada por

$$\Lambda\left(i, i^*\right) \propto \exp\left(-\left(i - i^*\right)^2 / 2\sigma^2\right). \tag{8}$$

El efecto que introduce la función de *función vecindad* es que la actualización de las neuronas sea local, siendo mas local cuanto menor sea el valor de σ.

El entrenamiento se realizó para tres valores de σ: 0,01, 0,5 y 5. Para cada uno de estos valores, se entreno a la red con 10, 100 y 10000. En todos los casos se utilizó $\eta = 0{,}01$. En la Figura 3 se observan los resultados del entrenamiento. Adicionalmente, se indica con un área sombreada la región correspondiente a la distribución de la entrada de la red y las posiciones iniciales de las neuronas de la red.

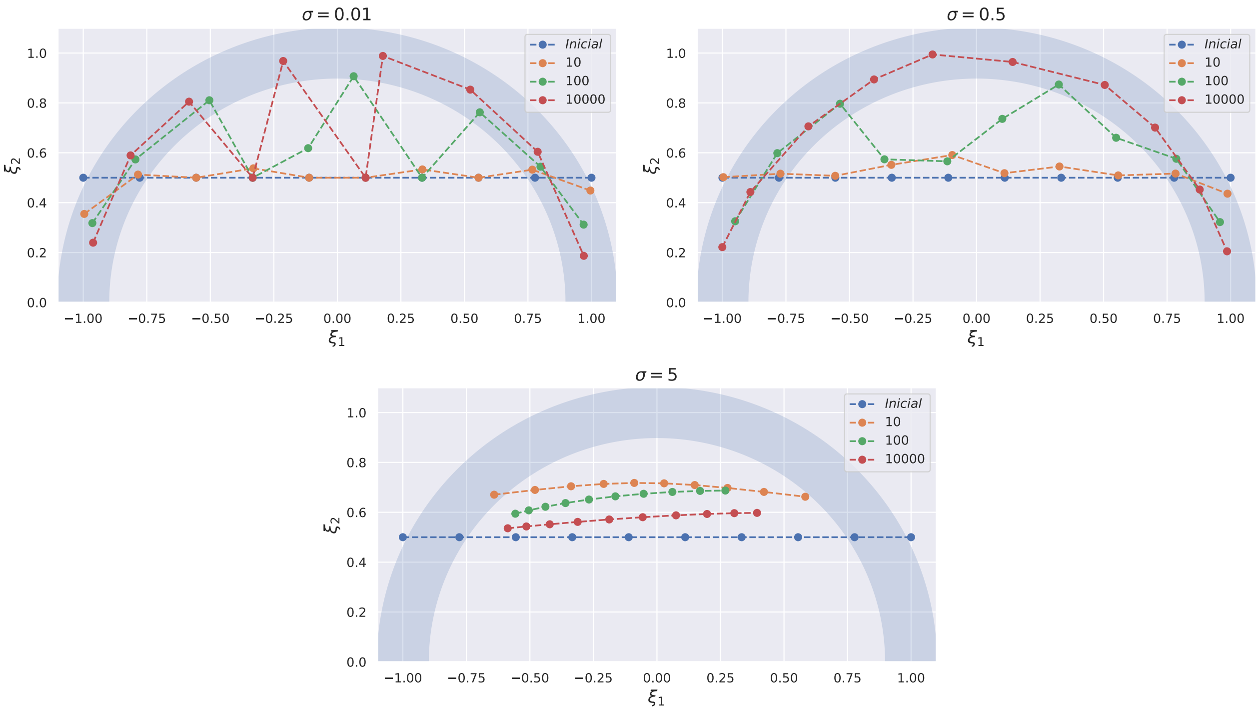

Figura 3: Se observa la disposición final de las neuronas para distintos valores de σ y distinta cantidad de entradas. Adicionalmente, se indica con un área sombreada la región correspondiente a la distribución de la entrada de la red y las posiciones iniciales de las neuronas.

Para $\sigma = 0{,}01$ las neuronas presentan dos comportamientos completamente diferentes. Una posibilidad es que la posición final de las neuronas se encuentre dentro del área sombreada, mientras que aquellas que no lo logran, se encuentran en zonas muy cercanas a su posición inicial. Esto se debe a que para un σ pequeño, las actualizaciones de los pesos son muy locales y solo la neurona ganadora es la que se ve modificada de forma significativa. Esto favorece que las neuronas ganadoras sean las que nuevamente serán modificadas en iteraciones posteriores (ya que sera mas probable que se encuentren cerca de la zona sombreada).

Con $\sigma = 5$ es el caso opuesto, en donde las modificaciones de los pesos globales, con lo cual prácticamente todas las neuronas se mueven de la misma manera.

Por último, se observa que $\sigma = 0{,}5$ es un equilibrio entre los dos comportamientos previamente descriptos y que, con la cantidad suficiente de iteraciones, la red aprende la forma de la distribución de la entrada sin problemas.