Redes Neuronales Artificiales
Práctica 4

1. Aprendizaje Hebbiano No Supervisado

Para resolver los siguientes ejercicios se debe programar el algoritmo de Aprendizaje Hebbiano No Supervisado que permita utilizar tanto la regla de Oja como la de Sanger. Utilizar para esto el lenguaje Matlab/Octave o el lenguaje Python junto las librerías NumPy y MatPlotLib.

1.1. Hiper-Rectángulo:

Dado un vector $A =< a_1, a_2, \ldots a_6 >$ en donde $\forall i : a_i \in \mathbb{N}$ y $\forall i, j : i \neq j \implies a_i \neq a_j$. Se define un conjunto de datos en donde cada caso $X =< x_1, x_2, \ldots x_6 >$ cumple con $x_i \in \mathcal{U}(-a_i, a_i)$. Es decir, cada caso pertenece a un hiper-rectángulo con todos sus lados de diferente tamaño según una distribución aleatoria uniforme en un intervalo dado por A.

Se pide resolver los siguientes puntos utilizando primero la regla de Oja y luego la de Sanger, de forma que sea posible comparar los resultados y extraer conclusiones.

1. Entrene una red de 6 entradas y 4 salidas presentando como entrada puntos de X. Realice varias corridas con distintos pesos iniciales.
2. Analice los vectores de pesos obtenidos. Verifique las propiedades vistas en las clases teóricas.
3. Teniendo en cuenta que la varianza de las proyecciones sobre los ejes cartesianos es $\frac{a_i^2}{3}$ en cada caso y que la matriz de covarianzas es diagonal (verifíquelo), ¿cuáles son las cuatro primeras componentes principales?
4. Calcule las salidas de la red para los puntos utilizados para el entrenamiento. Para cada salida, calcule la media y la varianza. Analice y justifique los resultados obtenidos.
5. ¿Qué ocurre si $a_i = a_j$ para algún par i, j?

1.2. Combinaciones Lineales:

Dadas 2 variables aleatorias de distribución uniforme $u_1, u_2 \in \mathcal{U}(-1, 1)$ y las siguientes matrices:

$$M_1 = \begin{bmatrix} 4 & 0 & 1/2 \\ 0 & 2 & 1/2 \end{bmatrix} \quad M_2 = \begin{bmatrix} 1 & 1/2 & 3/4 \\ 1/2 & 1 & 3/4 \end{bmatrix} \quad M_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1/2 & 1/2 & 1/2 \end{bmatrix}$$

Generar tres conjuntos de datos X_1, X_2 y X_3 dados por $X_i = [u_1, u_2]^\top \cdot M_i$ para entrenar varias redes hebbianas de 3 unidades de entrada y 2 de salida con las reglas de Oja y Sanger.

1. Comparar las matrices de pesos con las matrices M_i utilizadas para generar los datos de entrada.

2. Comparar los valores pertenecientes a X_i con los valores Y devueltos por la red.

3. Comparar los valores pertenecientes a X_i con los resultantes de hacer $Y \cdot W_i^\top$.

4. ¿Qué va a suceder si se utilizan los tres conjuntos de datos para entrenar la misma red?