

Aprendizaje no supervisado

Pablo Chehade

pablo.chehade@ib.edu.ar

Redes Neuronales, Instituto Balseiro, CNEA-UNCuyo, Bariloche, Argentina, 2023

EJERCICIO 1: NO CONTROLADO RESPECTO A NOTION

Se entrenó de manera no supervisada una red neuronal lineal de una sola capa con cuatro entradas y una salida. Los datos de entrada presentan una distribución gaussiana con matriz de correlación Σ

$$\Sigma = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

El autovector asociado al mayor autovalor de Σ es $vecv = \frac{1}{2}(1, 1, 1, 1)$.

A partir de un vector de pesos inicial $vecw = (w_1, w_2, w_3, w_4)$, donde cada componente w_j adopta un valor aleatorio no mayor a 0,01, se aplicó la regla de Oja:

$$\Delta w_j = \eta(\xi_j - Vw_j)$$

Aquí, η representa la tasa de aprendizaje, V es la salida de la red y ξ_j es la componente j del dato de entrada ξ .

Bajo estas condiciones, se espera que el vector de pesos $vecw$ tienda al autovector $vecv$.

En la figura [1], se ilustra la evolución del módulo de los pesos, las modificaciones Δw_j y la diferencia entre w_j y v_j , para $\eta = 0,001$. Tras un período transitorio, se observa una tendencia en las componentes de $vecw$ hacia el valor 0,5, indicando convergencia hacia \vec{v} . Dependiendo de las condiciones iniciales, el vector \vec{w} apunta en la dirección $+\vec{v}$ o $-\vec{v}$. Debido a esto se graficó el módulo de cada componente. Posteriormente, en estado estacionario, los pesos continúan modificándose. Por ello, en la figura [2], se procesan los datos anteriores realizando un promedio en cada paso de tiempo de los 200 pasos adyacentes. Esto resulta en un comportamiento más suave y una reducción notable en las variaciones de Δw_j de hasta un orden de magnitud.

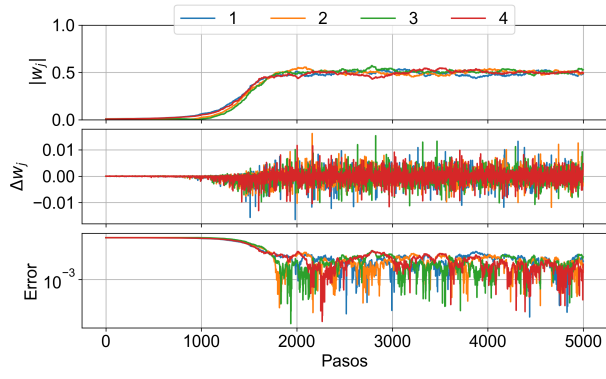
Por otro lado, en la figura [3], se evolucionó el sistema con $\eta = 0,01$. El sistema muestra una convergencia más rápida y variaciones Δw_j de mayor magnitud. Además, se determinó que existe un valor superior límite para η , más allá del cual el sistema tiende a diverger.

EJERCICIO 2

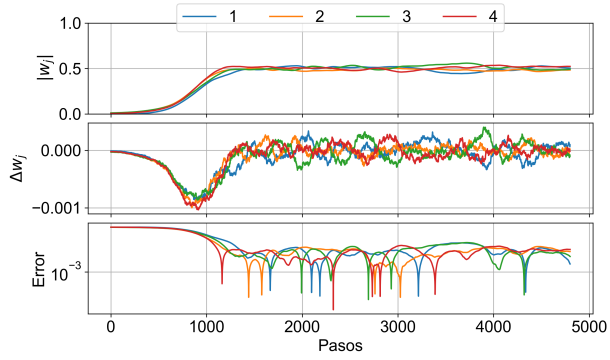
APÉNDICE

A continuación se desarrolla el código empleado durante este trabajo implementado en Python.

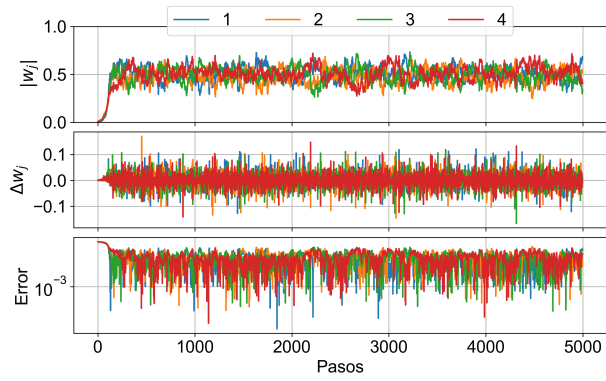
```
1 #Import libraries
2 import numpy as np
3 import matplotlib
4 import matplotlib.pyplot as plt
5 import tensorflow as tf
```



(a)



(b)



(c)

Figura 1: Three simple graphs