



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II

Aprendizaje Hebbiano No Supervisado

Redes Neuronales
Primer Cuatrimestre 2020

Integrante	LU	Correo electrónico
Ilan Olkies	250/17	ilanolkies@gmail.com
Joaquín Naftaly	816/17	joaquin.naftaly@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

1. Introducción

En este trabajo presentamos dos modelos de *aprendizaje hebbiano no supervisado* para reducir la dimensionalidad de documentos con descripciones de texto correspondientes a compañías Brasileñas clasificadas en nueve categorías distintas.

Para este problema contamos con un dataset de 900 instancias clasificadas en 9 categorías, y cada una con 850 atributos correspondientes a las frecuencias de palabras en el documento. El objetivo es reducir la dimensionalidad de las instancias a 9 dimensiones.

En este trabajo experimentamos con dos reglas de *aprendizaje hebbiano no supervisado*:

1. Oja Generalizada
2. Sanger

2. Modelo de aprendizaje Hebbiano

A diferencia del *aprendizaje supervisado* no tenemos datos objetivo para comparar la respuesta de la red y así calcular el error y corregir los pesos. En este caso la red misma se encarga de encontrar patrones en los datos. El aprendizaje de la red se basa en el Postulado de Hebb: "*La conexión entre dos neuronas se refuerza si las mismas se activan juntas reiteradas veces*". De este postulado se deduce la siguiente regla de aprendizaje:

$$\Delta W_{ij} = n * X_i * Y_j \quad (1)$$

En la ecuación 1 n es el *learning rate*, X son los datos de entrada e Y es la respuesta de la red. Esta regla tiene dos problemas. El primero es que, al final del entrenamiento, todas las M neuronas de salida terminan aprendiendo lo mismo. Y el segundo es que los pesos W crecen indefinidamente durante el entrenamiento haciendo que la red sea poco estable.

2.1. Oja Generalizado

La regla de aprendizaje propuesta por *Oja* para una red con M neuronas de salida es la siguiente:

$$\Delta W_{ij} = n * (X_i - \bar{X}_i) * Y_j \quad , \quad \bar{X}_i = \sum_k^M W_{ik} * Y_k \quad (2)$$

Esta regla soluciona los problemas de la regla de *Hebb*. Los pesos W se mantienen normalizados y cada neurona de salida aprende características distintas de los datos.

2.2. Sanger

La regla de *Oja Generalizada* hace que los pesos W terminen alineados en un subespacio equivalente al de las primeras M componentes principales. Pero *Sanger* propuso una regla de aprendizaje en la que la i -ésima neurona de salida aprende la i -ésima componente principal de los datos. Esta regla es la siguiente:

$$\Delta W_{ij} = n * (X_i - \bar{X}_i) * Y_j \quad , \quad \bar{X}_i = \sum_k^j W_{ik} * Y_k \quad (3)$$

Notar que para que este modelo funcione correctamente los datos de entrada deben estar normalizados. Esto quiere decir que los datos tienen media 0 y varianza 1.

3. Experimentación

Para evaluar las dos reglas de aprendizaje propuestas contamos con un dataset de 900 instancias, los documentos representados en Bag of Words. Del dataset tomamos las primeras 700 instancias para entrenar el modelo y las siguientes 200 para testarlo. Una vez definidos los parametros del modelo, lo entrenamos con las 900 instancias juntas.

El entrenamiento del modelo consta en aplicar la matriz de aprendizaje de cada regla repetidas veces a la matriz de train, hasta obtener una matriz lo suficientemente ortogonal. Luego, para evaluar al conjunto de test sobre el modelo entrenado, se activa la red simplemente multiplicando las entradas por la matriz calculada.

Para ambas reglas de aprendizaje evaluamos los resultados modificando tres parámetros principales del problema: la cantidad máxima de épocas, la mínima ortogonalidad aceptable y el *learning rate* que permite aplicar el aprendizaje de forma gradual.

El entrenamiento de ambos modelos tenia un comportamiento particular. Para cada modelo, la ortogonalidad quedaba siempre oscilando en algún punto. Bajando el *learning rate* el algoritmo lograba bajar de ese valor, por lo que decidimos aplicar el *learning rate* en dos etapas: una primera con un *learning rate* alto para llegar mas rápido a ese valor y un segundo *learning rating* más pequeño para romper esa oscilación y seguir reduciendo el error.

3.1. Oja

Para el modelo entrenado con la regla de aprendizaje de Oja usamos un *learning rate* de 0.00001. Cuando la ortogonalidad esta entre 0.017 y 0.008 aplicamos la regla de la reducción del *learning rate*: se divide en 10 cada 100 épocas que el aprendizaje esté ocilando entre los limites definidos.

Con los parámetros elegidos logramos llevar la ortogonalidad a 0.002410504514176649. El gráfico del modelo entrenado muestra como los elementos quedaron categorizados casi unívocamente.

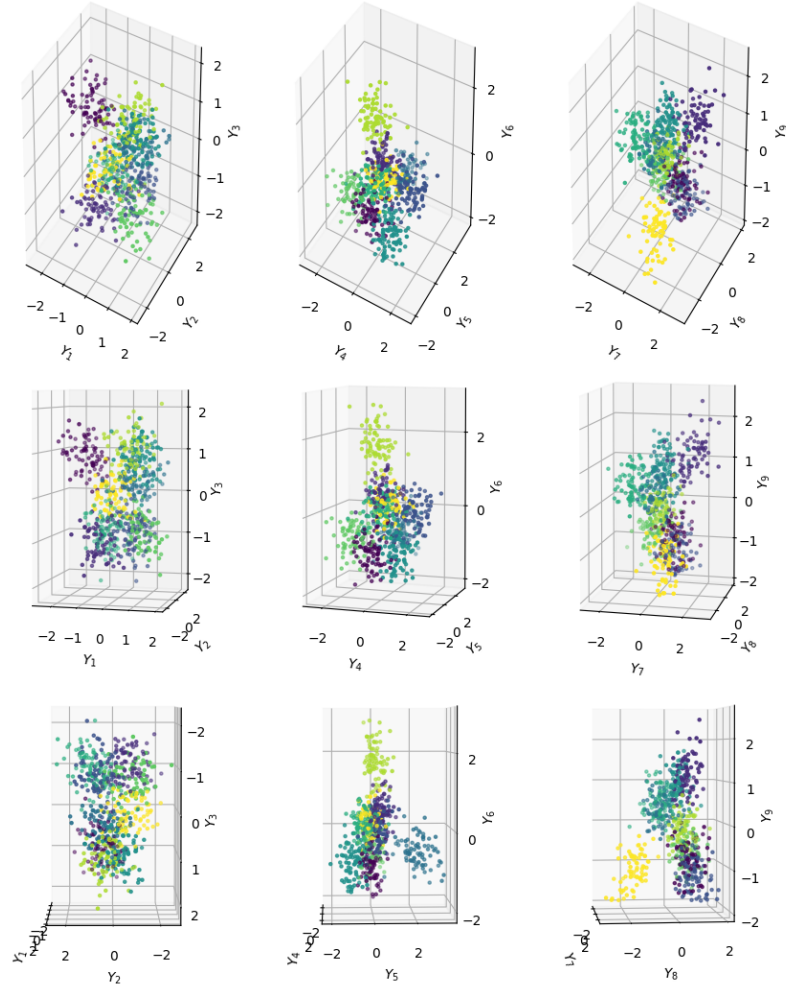


Figura 1: Entrenamiento con regla de Oja

El conjunto de test da un resultado bueno también, los documentos se acomodaron según su categoría.

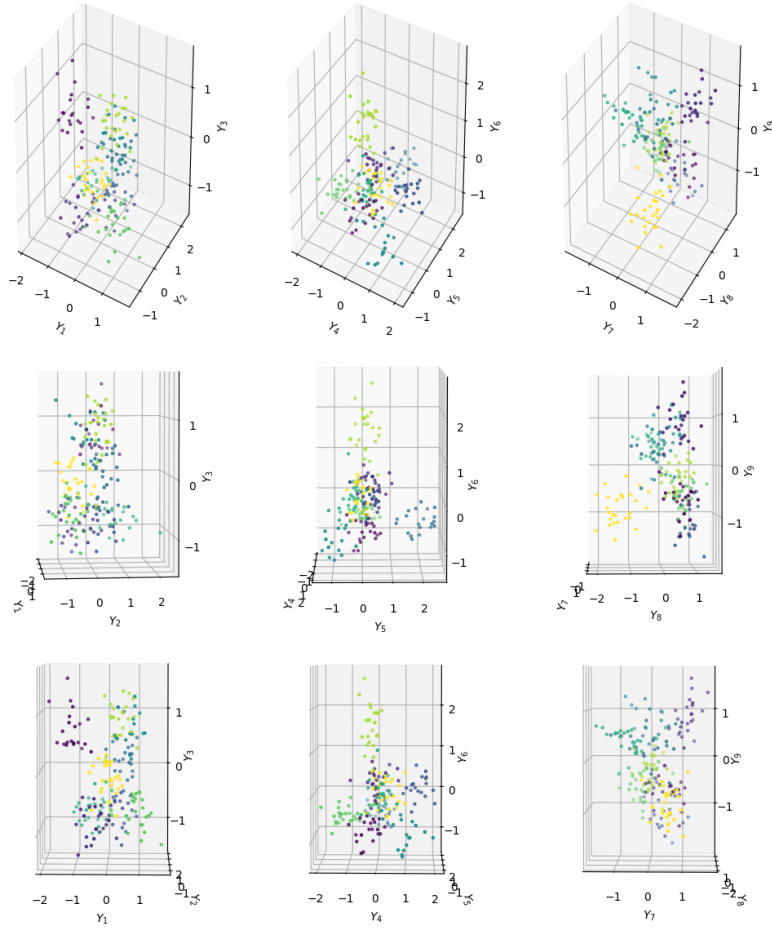


Figura 2: Test con regla de Oja

En este gráfico podemos ver como la ortogonalidad del modelo se acerca a 0 por cada época del entrenamiento.

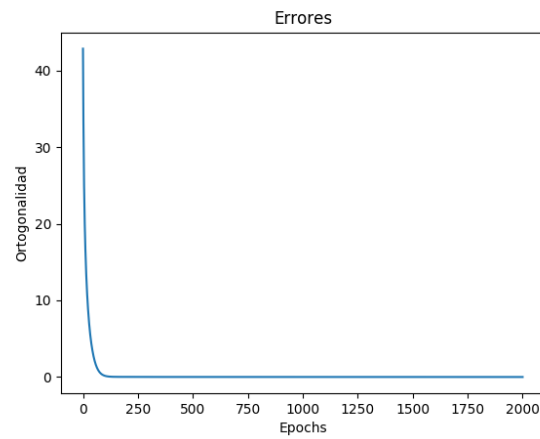


Figura 3: Error en el entrenamiento con regla de Sanger

Finalmente entrenamos el modelo con todo el dataset y los mismos parámetros resultantes de la experi-

mentación. Obtenemos una ortogonalidad de 0.0016511 y el siguiente modelo:

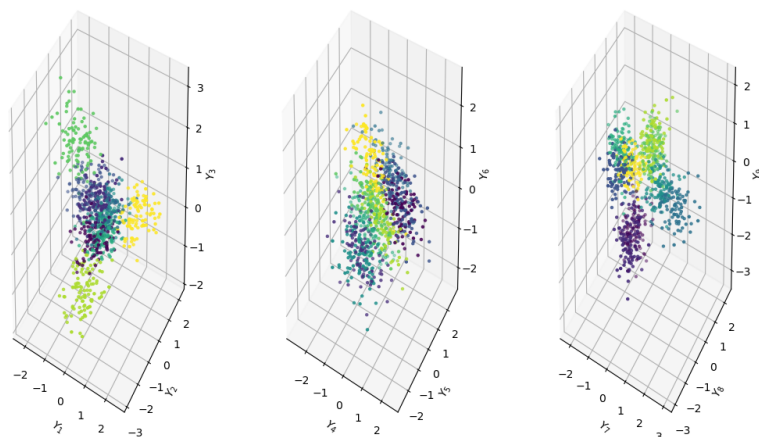


Figura 4: Modelo entrenado con regla de Oja

3.2. Sanger

La regla de aprendizaje de Sanger, al igual que la de Oja, tiene un punto de quiebre entre el 0.038 y 0.0379. Cada 100 iteraciones dentro de ese rango el learning rate se divide en 10. En este caso arrancamos con un *learning rate* de 0.00001, y una ortogonalidad mínima aceptada de 0.001. La ortogonalidad alcanzada en este entrenamiento fue de 0.04941480901417489.

En el caso de *Sanger* los ejes de los gráficos representan a las componentes principales. Podemos percibir que usando el modelo de Sanger los resultados se ven aún más agrupados por categoría que en el de Oja.

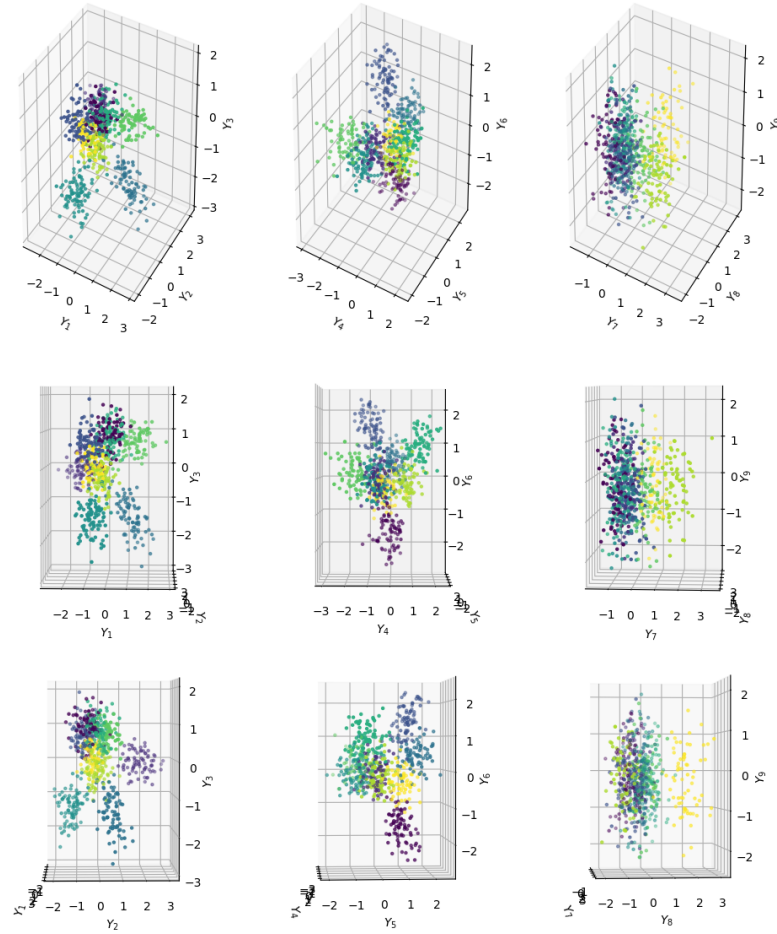


Figura 5: Entrenamiento con regla de Sanger

Luego de procesar la activación, los resultados de las categorías se pueden apreciar en estos gráficos

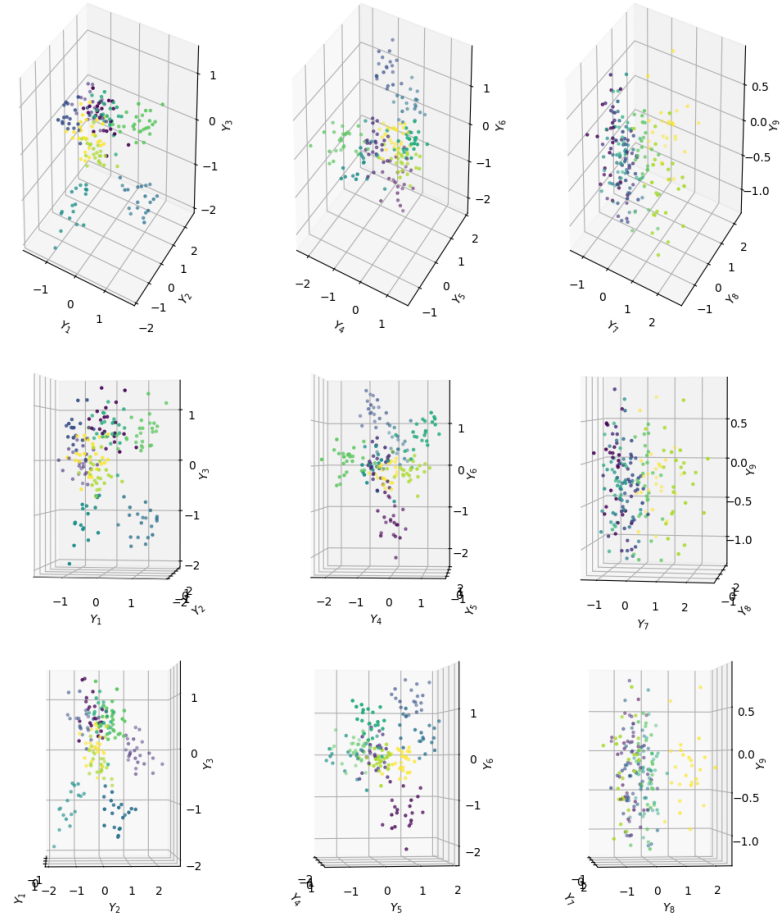


Figura 6: Test con regla de Sanger

Como para el algoritmo anterior, este gráfico muestra la ortogonalidad en cada época.

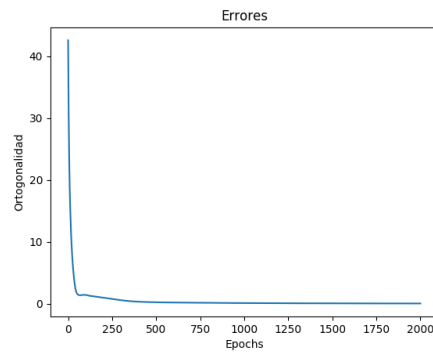


Figura 7: Error en el entrenamiento con regla de Sanger

Por ultimo, entrenando con todos los datos de entrada la ortogonalidad resultante es 0.001648.

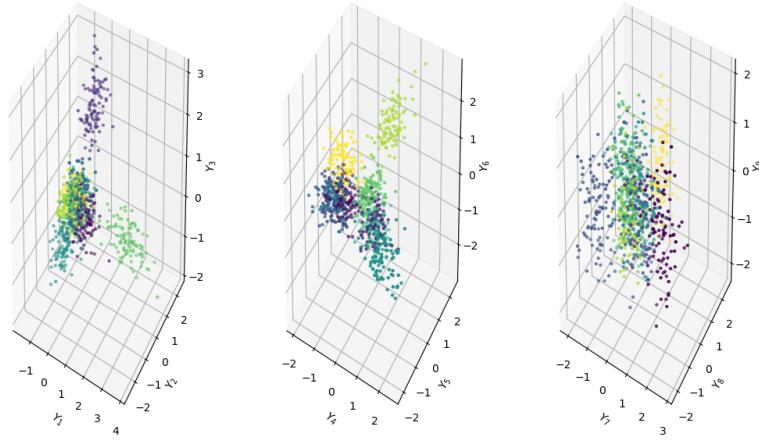


Figura 8: Modelo entrenado con regla de Sanger

4. Conclusión

La evaluación realizada sobre los resultados de la ejecución de los modelos es simplemente una categorización visual. Esto facilita el entrenamiento de una red neuronal para clasificar los documentos según su categoría ya que la dimensión de los datos es mucho menor y además están agrupados según su categoría.