

Trabajo Práctico 2: Redes Neuronales Artificiales

L. Alvarez, P. Bordon y D. Santos

31 de Octubre de 2016

Índice

1. Descripción del problema	3
2. Ejercicio 1: Reducción de dimensiones	4
2.1. Introducción	4
2.2. Análisis de la Red	4
2.3. Procesamiento de Datos	4
2.4. Experimentación y Resultados	4
2.4.1. Algoritmo de Oja	4
2.4.2. Algoritmo de Sanger	9
2.5. Conclusiones	12
3. Mapeo de Características	13
3.1. Implementación del Algoritmo	13
3.2. Preprocesamiento de Datos	13
3.3. Dimensiones Adecuadas	13
3.3.1. Dimensión 10 x 10	14
3.3.2. Dimensión 20 x 20	14
3.3.3. Dimensión 30 x 30	15
3.3.4. Conclusión sobre la Dimensiones	16
3.4. Vecindad, Learning Rate y Sigma	16
3.4.1. Buscando la Combinación Ideal	16
3.4.2. Conclusiones de los Parámetros	17
3.5. Cotas	17
3.6. Entrenamiento	17
3.6.1. 50 Datos de Entrada	17
3.6.2. 200 Datos de Entrada	21
3.6.3. 800 Datos de Entrada	24
3.6.4. El mejor mapa	27
3.6.5. Futuros Entrenamientos	28
3.7. Detalles de Implementación	29
3.8. Detalles de Resultados	29
3.9. Conclusiones	30

1. Descripción del problema

Se cuenta con documentos de descripción de empresas. Dichos documentos se encuentran preprocesados, de manera tal que se tiene un dataset que especifica qué cantidad de veces aparece cada palabra en cada documento. Este formato es conocido como Bolsa de Palabras (Bag-Of-Words). El problema a resolver será diseñar y entrenar una red neuronal mediante métodos de aprendizaje hebbiano no supervisado para que pueda clasificar automáticamente los documentos según categoría. También se busca que esta red pueda generalizar correctamente y responda de manera certera a nuevos documentos.

Se utilizaron dos subparadigmas clásicos de aprendizaje no supervisado: aprendizaje hebbiano y aprendizaje competitivo. Se trabajó en un modelo de reducción de dimensiones y otro de mapeo de características autoorganizado para su aplicación sobre las datos propuestos. En el primer caso se planteó una reducción a tres dimensiones utilizando los algoritmos de Oja y Sanger. En el segundo se planteó un modelo de autoorganización a partir del algoritmo de Kohonen. Distintas configuraciones de parámetros fueron estudiadas en cada caso. Se presentan aquí los resultados obtenidos.

2. Ejercicio 1: Reducción de dimensiones

2.1. Introducción

Para el primer ejercicio, queremos reducir la dimensionalidad de la entrada, debido a que es muy grande, a 3 valores. Para esto vamos a diseñar una red neuronal y la entrenaremos mediante aprendizaje hebbiano no supervisado aplicando la regla de oja y la regla de sanger. Luego analizaremos y compararemos sus resultados.

2.2. Análisis de la Red

En un comienzo, trabajamos con una red que contaba con 10 o 15 neuronas de salida para no reducir drásticamente su dimensión. Al final el entrenamiento, reducíamos la salida a 3 valores y obteníamos lo buscado. Luego de varias pruebas y cambios determinamos que no ganábamos nada tomando esta determinación, por el contrario, se aumentaba la complejidad del algoritmo y se perdía tiempo revisando la ortogonalidad de las 10 columnas de la matriz de pesos, mientras que los resultados obtenidos no eran mejores. Por esto se determinó manejar desde un inicio 3 dimensiones de salida.

Vale aclarar que las reglas de oja y sanger son formas de calcular el cambio que se debe hacer a la matriz de pesos para entrenarla. Por esto, ambos modelos son iguales, solo que se aplican distintos cálculos para cada tipo de regla.

2.3. Procesamiento de Datos

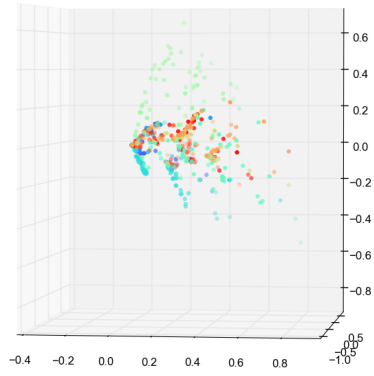
Sabemos que los valores de entrada son enteros positivos, ya que determinan la cantidad de apariciones de una palabra en un texto. Y al estar la entrada preprocesada y sin las palabras mas comunes (con mas apariciones), los valores de entrada se encuentran acotados. Podríamos no haber implementado ningún tipo de preprocesamiento, pero preferimos estandarizar la entrada, de manera que los atributos presenten media cero y varianza uno, moviendose en un rango de valores similar.

2.4. Experimentación y Resultados

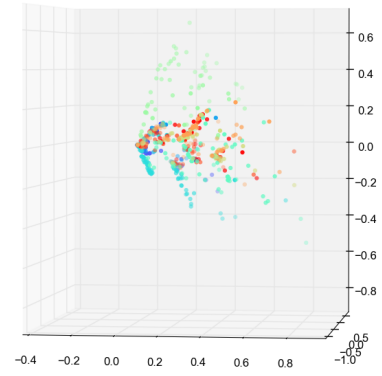
Habiendo fijado el número de neuronas de salida a tres unidades, pasamos a implementar los dos algoritmos propuestos. En primera instancia trabajamos en una fase de entrenamiento, donde el parámetro de interés a optimizar es el coeficiente de aprendizaje, que bien puede ser función del número de época. Tanto la condición de ortonormalidad ($\|W^T \cdot W - I\| = 0$) como la subsiguiente validación son las herramientas para determinar su valor óptimo. Comenzamos con un coeficiente de aprendizaje constante, siguiendo con funciones del tipo $\frac{1}{\text{epoca}^\alpha}$, con alfa entre 0 y 1. Probando estas distintas opciones encontramos que si bien no existían diferencias significativas, un coeficiente del tipo $\frac{1}{\text{epoca}^{\frac{1}{2}}}$ presentaba los mejores resultados.

2.4.1. Algoritmo de Oja

A continuación se pueden observar los resultados obtenidos utilizando el algoritmo de Oja separando el set de datos de entrada en 70 % - 30 %, para el coeficiente de aprendizaje adaptativo propuesto. Las figuras 1 y 2 presentan los datos en el espacio 3d de salida obtenido por el algoritmo. Las figuras 1a y 2a corresponden a datos de entrenamiento, mientras que las figuras 1b y 2b corresponden a datos de validación.

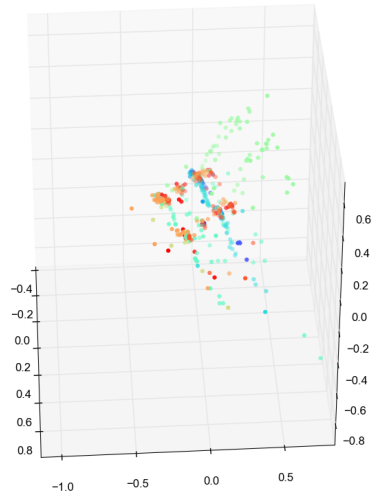


(a) Entrenamiento

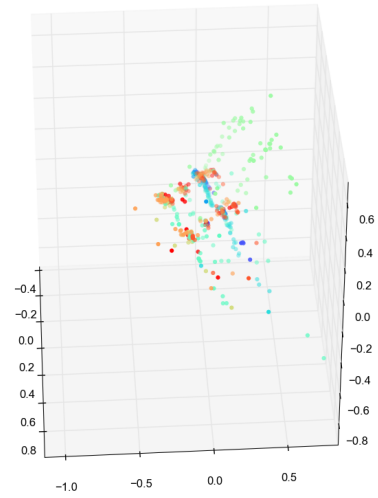


(b) Validacion

Figura 1: Vista 1 en 3d de resultados de validación y entrenamiento



(a) Entrenamiento



(b) Validacion

Figura 2: Vista 2 en 3d de resultados de validación y entrenamiento

Se puede observar que la validación es consistente con la clasificación obtenida en la etapa de entrenamiento,

dando cuenta de la validez del procedimiento realizado. También se puede observar en el gráfico claramente la separación entre dos de las categorías, la verde y azul, cosa que no ocurre para las categorías restantes. Para poder discriminar sobre estas últimas realizamos graficos 2d (figura 3), proyectando el gráfico 3d sobre tres planos distintos, tanto para los datos de entrenamiento 3 como para los de validación 4.

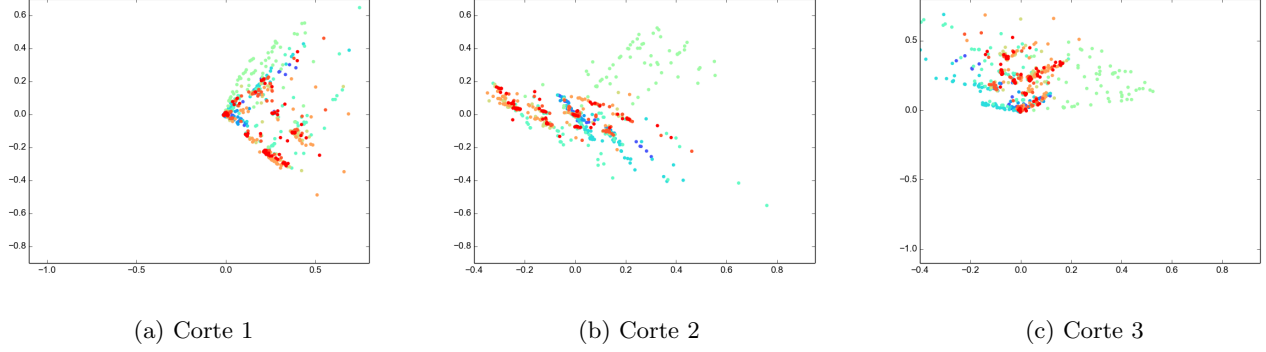


Figura 3: Cortes del gráfico de los datos de entrenamiento

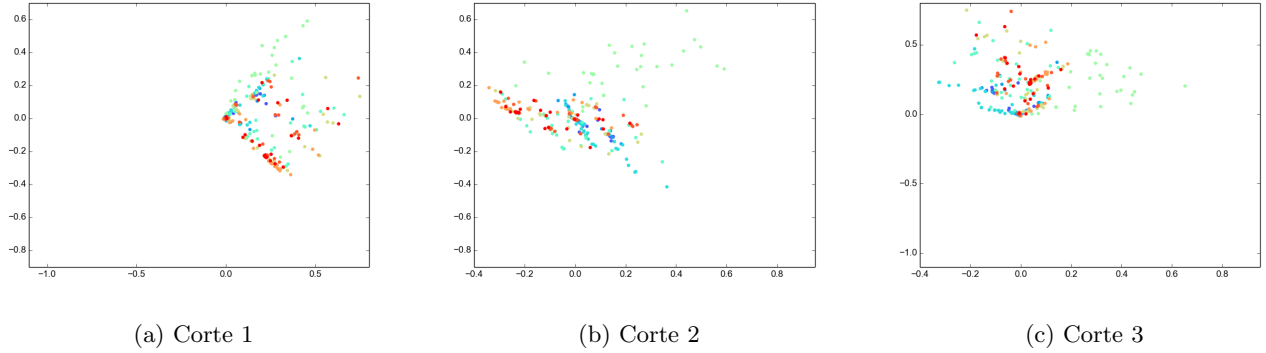
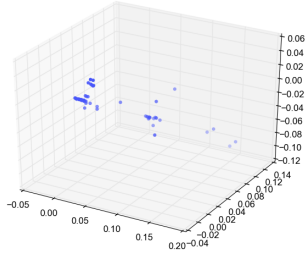


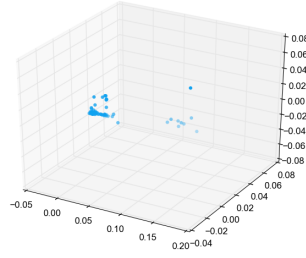
Figura 4: Cortes del gráfico de los datos de validación

Estos gráficos permiten observar mejor las similitudes entre los resultados obtenidos con los datos de entrenamiento y los obtenidos con los datos de validación. No se distinguen resultados tan errados como para ser detectados a simple vista. Esto nos deja la impresión de que, a pesar de que las categorías no se separen y agrupen como esperábamos, la red generaliza bastante bien y es consistente en cuanto a la ubicación de las categorías.

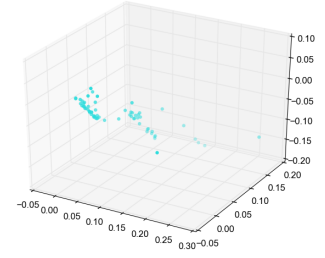
A continuación realizamos un gráfico 3D para cada categoría en la etapa de entrenamiento, con la finalidad de poder observar su distribución individualmente, ya que en las figuras ?? se encuentran algo superpuestas. Los resultados se observan en las figuras 5.



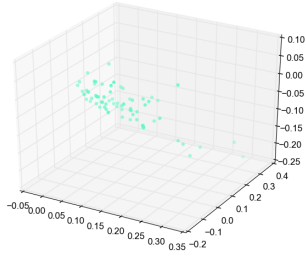
(a) Categoría 1



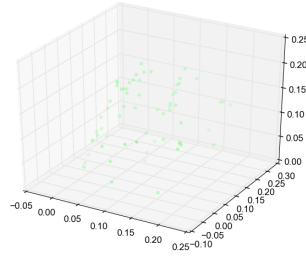
(b) Categoría 2



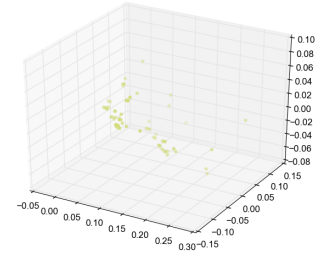
(c) Categoría 3



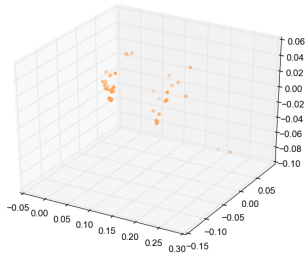
(d) Categoría 4



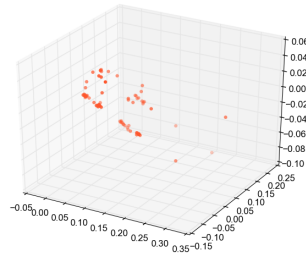
(e) Categoría 5



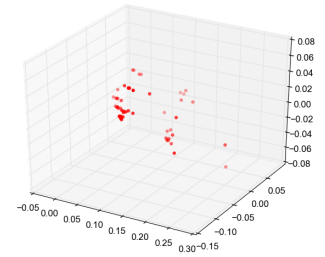
(f) Categoría 6



(g) Categoría 7



(h) Categoría 8



(i) Categoría 9

Figura 5: Gráficos de resultados diferenciados por categoría

Como observamos que algunas categorías se encontraban algo dispersas, calculamos la varianza promedio de los datos de cada categoría para ver si nos dan alguna señal de por qué puede estar ocurriendo esto. Los resultados se pueden observar en la siguiente figura 6.

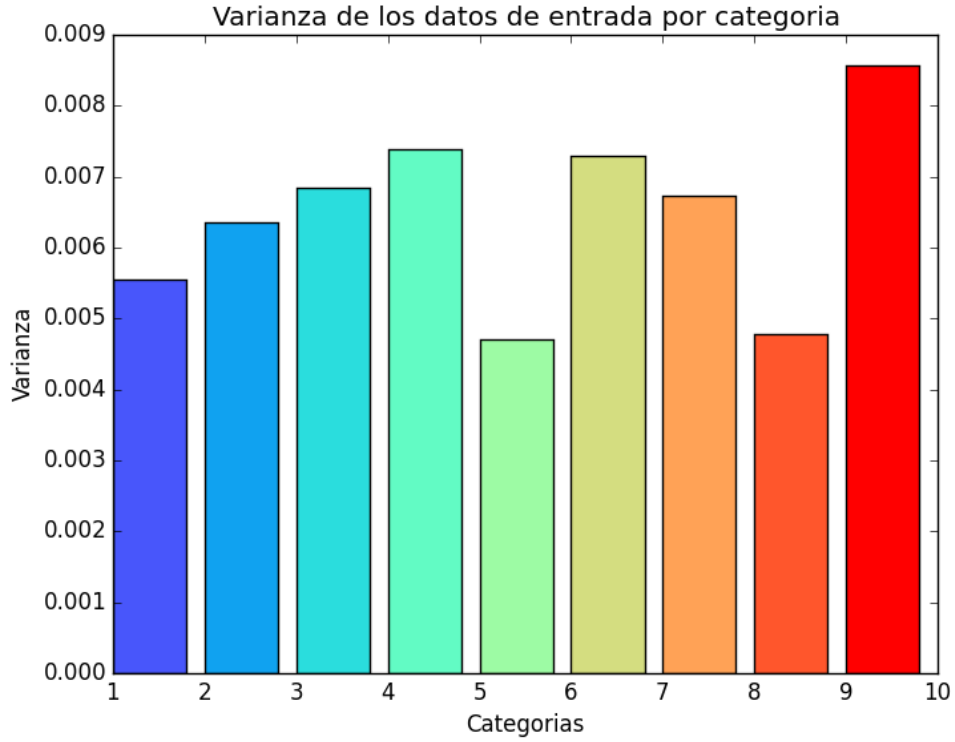
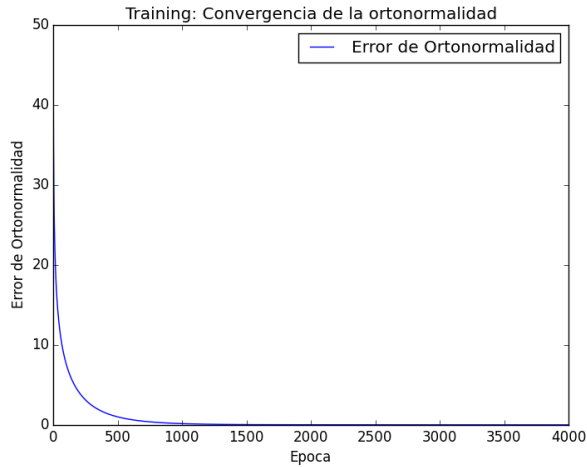
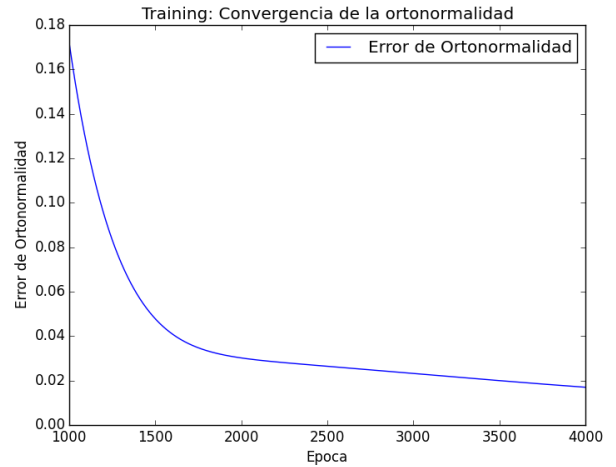


Figura 6: Varianza de los datos de entrada por categoría

En la figura 7, se puede observar como se satisface la condición de ortonormalidad, en función del número de época. Se aprecia que a partir de las 1000 épocas la norma se hace menor al 0.18 y continúa decreciendo fuértemente hasta valores por debajo de 0.02.



(a) Todas las épocas



(b) A partir de la época 1000

Figura 7: Ortonormalidad de la matriz de pesos en función de las épocas.

2.4.2. Algoritmo de Sanger

Se realizó una análisis similar al caso anterior, utilizando el mismo porcentaje de datos de entrenamiento - validación y el mismo coeficiente de aprendizaje adaptativo. En la figuras 8 y 9 se puede observar vistas 3d (componentes principales obtenidos por el algoritmo) de los datos de entrenamiento (figuras 8a, 9a) y validacion (figuras 8b, 9b). Al igual que en el caso anterior, obtenemos una buena clasificación sobre los datos de validación.

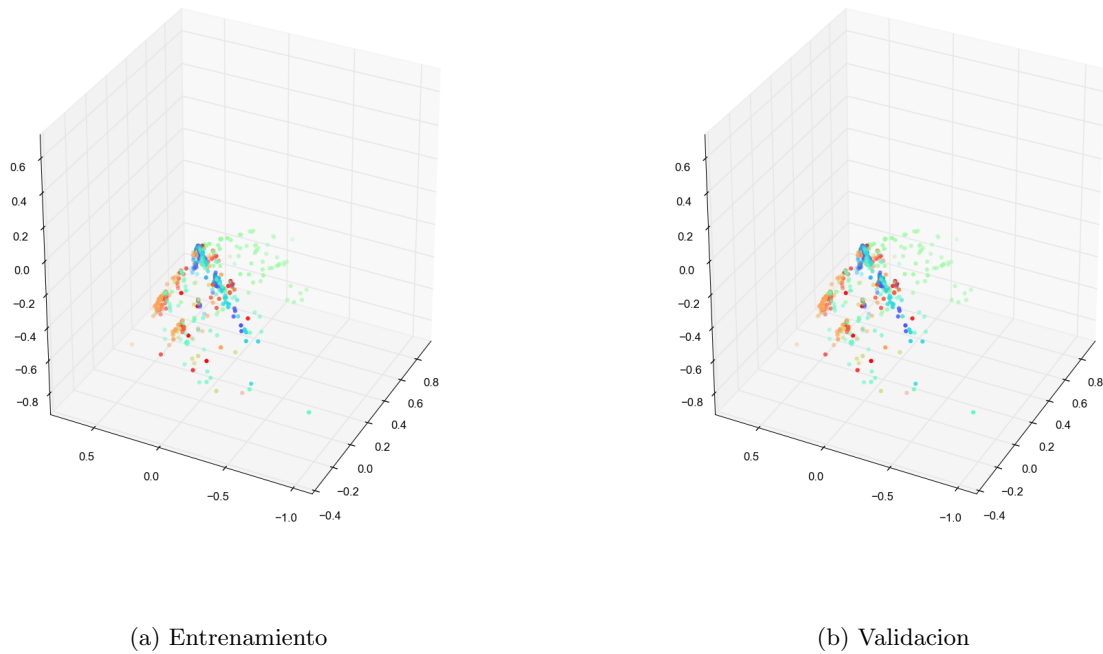
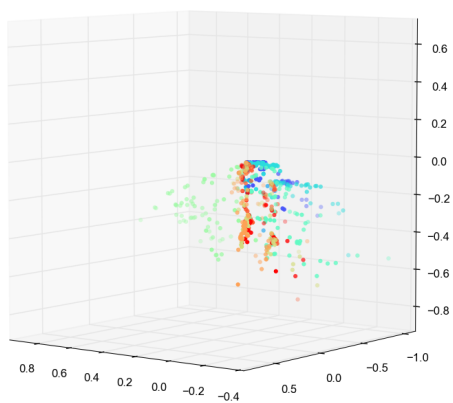
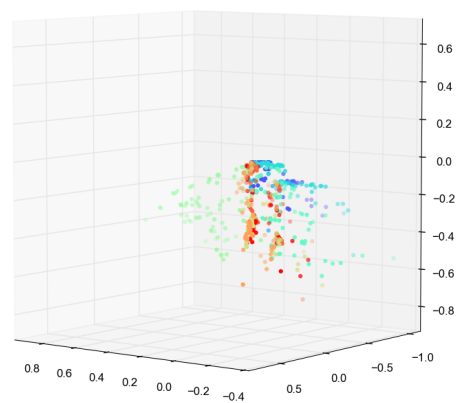


Figura 8: Vista 1 en 3d de resultados de validación y entrenamiento



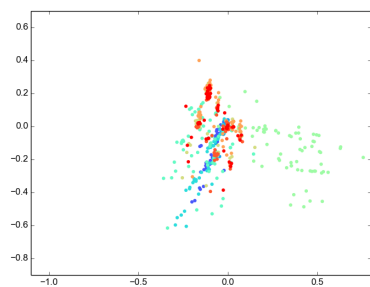
(a) Entrenamiento



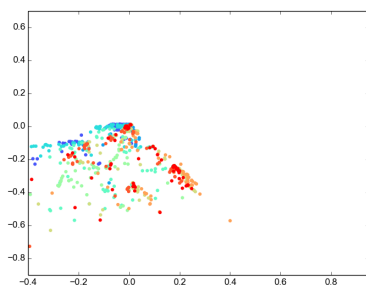
(b) Validacion

Figura 9: Vista 2 en 3d de resultados de validación y entrenamiento

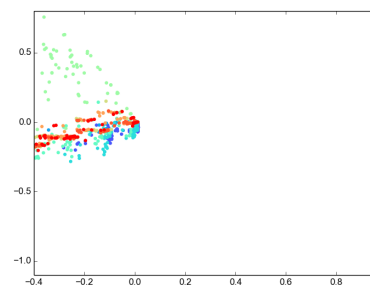
proyectando estos gráficos 3d sobre distintos planos, obtenemos las siguientes figuras, tanto para entrenamiento figuras 10 como para validación figura 11



(a) Corte 1

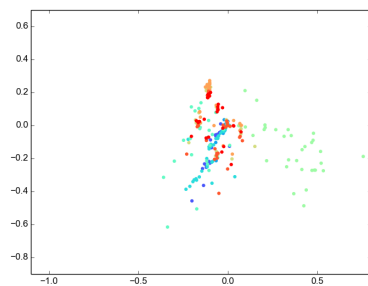


(b) Corte 2

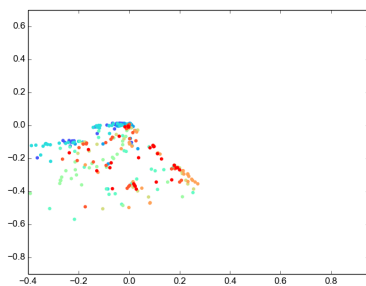


(c) Corte 3

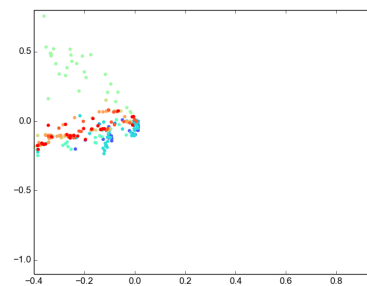
Figura 10: Cortes del gráfico de los datos de entrenamiento



(a) Corte 1



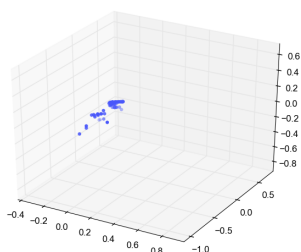
(b) Corte 2



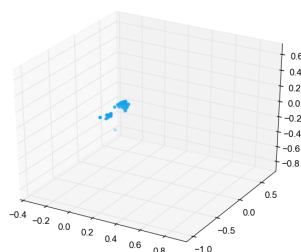
(c) Corte 3

Figura 11: Cortes del gráfico de los datos de validación

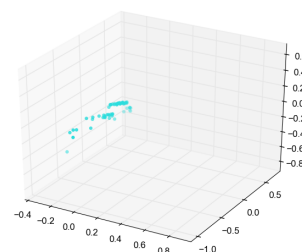
A continuación podemos observar los resultados de entrenamiento en gráficos 3d para las distintas categorías, con el fin de observar más claramente como se distribuyen los mismos (figuras ??).



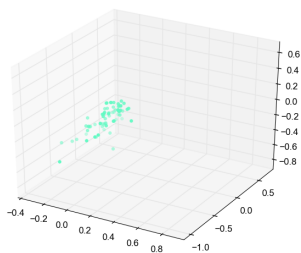
(a) Categoría 1



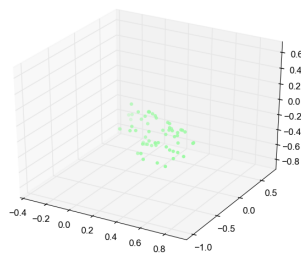
(b) Categoría 2



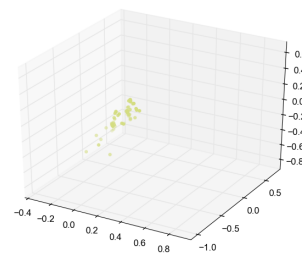
(c) Categoría 3



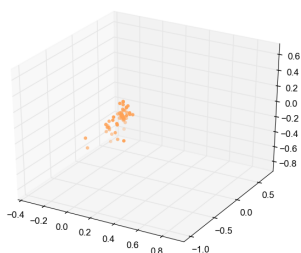
(d) Categoría 4



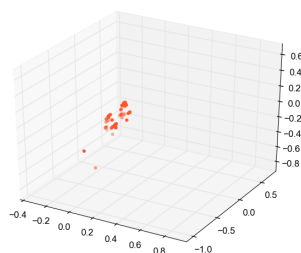
(e) Categoría 5



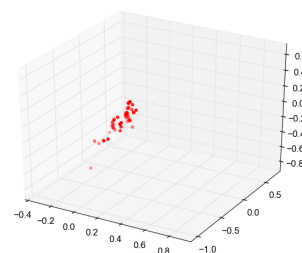
(f) Categoría 6



(g) Categoría 7



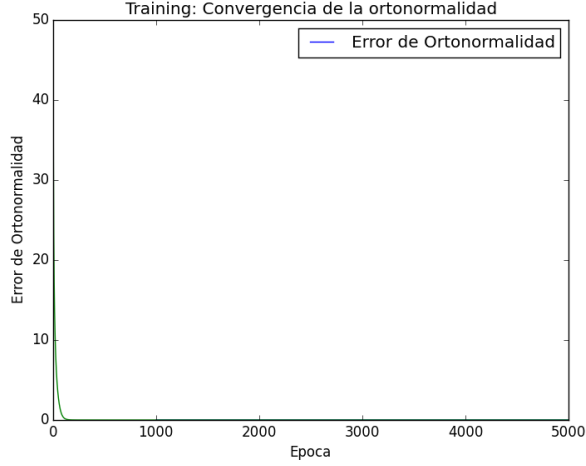
(h) Categoría 8



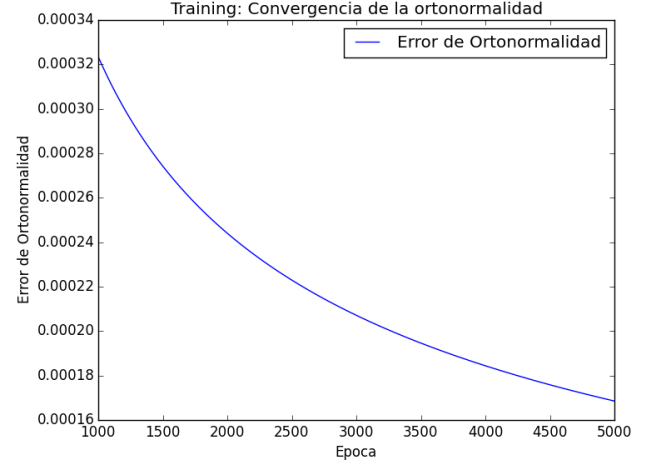
(i) Categoría 9

Cabe destacar que los datos de entrada son los mismos tanto para Oja como para Sanger. Por lo que la varianza observada en la figura 6 también aplica para esta regla de aprendizaje y debemos tenerla en cuenta a la hora de comprender la dispersión de los datos de las categorías en los gráficos obtenidos.

En las figuras 13, se puede observar como se satisface la condición de ortonormalidad, en función del número de época. Se aprecia que a partir de las 1000 épocas la norma se hace menor al 0.0004, un valor muy bajo, y continúa decreciendo.



(a) Todas las épocas



(b) A partir de la época 1000

Figura 13: Ortonormalidad de la matriz de pesos en función de las épocas.

2.5. Conclusiones

Se implementaron dos algoritmos para reducir la dimensionalidad del set de datos de entrada. Tanto el algoritmo de Sanger como el de Oja nos permiten obtener una clasificación que es consistente con los datos de validación. La diferencia entre los algoritmos radica en que el de Sanger nos permite obtener las componentes principales, mientras que el de Oja nos da un subespacio generado por estos, pero no nos da explícitamente las direcciones. Contrario a lo esperado, no pudimos visualizar esta diferencia.