DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

Aprendizaje No Supervisado

Redes Neuronales

Integrante	LU	Correo electrónico
Bonet, Felipe		fpbonet@gmail.com
Martínez, Federico		fedomartinez@hotmail.com
Avendano, Demian		demian.avendano@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

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

Índice

1. Introducción

En este documento se realizan las actividades propuestas en el TP 2, actividades relacionadas con la implementación de dos problemas de clasificación. El primero de ellos relacionado a la reducción de dimensionalidad de un set de datos. El segundo, confección de mapas autoorganizados. El objetivo del trabajo es desarrollar redes neuronales que propongan soluciones a ambos.

1.1. Introducción al problema

Las redes neuronales son modelos computacionales, en los que se intenta emular el funcionamiento fisiológico de un conjunto de neuronas biológicas, interconectadas, con el fin de lograr predicciones a partir de un conjunto de datos similares, presentados previamente. Para ello se modelan, en cada unidad de procesamiento, características que tienen que ver con las condiciones de propagación de señales electroquímicas. Estas condiciones se describen y modelan a partir de observaciones de sobre cómo es transmitida información entre una neurona y otra (o sobre si), y sobre como se encuentran interconectadas.

La suma de las interacciones entre estas unidades modeladas en una topología dada, genera propiedades emergentes que permiten resolver cierto tipos de problemas (en el caso de este trabajo, problemas de clasificación de elementos). Para intentar resolver estos problemas utilizando redes neuronales, es necesario recurrir a diversas técnicas para el ajuste de las variables de la red, y en muchos casos se requiere un paso de preprocesamiento de los datos.

En ambos casos utilizamos redes de aprendizaje no supervisado. Para el primer problema, la técnica utilizada fue aprendizaje hebbiano. Para el segundo, redes autoorganizadas.

Los valores de entrada fueron clasificaciones de empresas brasileras en base a 850 atributos, originalmente derivados de una descripción en palabras. La idea final del trabajo fue poder predecir estas clasificaciones con redes de aprendizaje no supervisado y poder entrenarlas para lograrlo de la manera más precisa posible.

1.2. Entrega

1.3. Requerimientos

- Intérprete python 2.7.
- Librerías estandar, librería *matplotlib* y *numpy*.
- Archivo CSV con uno de los dos formatos propuestos en el TP.

1.4. Modo de uso y opciones

Para usar este programa, se deben ejecutar el archivo **script.py**, el cual contiene todas las opciones de ejecución. Para ello, tipear por consola :

```
$python script.py N args
```

donde `N` es el número de ejercicio y `args` son los argumentos optativos. Es obligatorio proveer el número de ejercicio, ya que se parsean los inputs de forma distinta. A su vez, cambian los métodos de cálculo de eficacia de la red.

Las opciones disponibles son:

1.4.1. Opciones

1.5. Archivos

Para la resolución del trabajo, fue necesario desarrollar en primer lugar un parser de los datos de entrada. El código de las funciones de parsing esta en **script.py**. Allí se encuentran la inicialización del programa, junto con el parseo de los parámetros, su normalización y el llamado a las funciones y métodos de la red.

El código propiamente de la red se encuentra en **network.py**. Allí están todas las funciones de los procesos de inicialización, computación de pesos y resultados.

2. Resultados

En esta sección incluiremos los resultados de la experimentación que realizamos con las redes desarrolladas.

La idea general de los experimentos es intentar medir la performance de cada red observando, en el primer problema, la distribución espacial de los clusters de empresas y en el segundo, la calidad del mapa generado.

Presentamos en primer lugar los resultados obtenidos del problema de reducción de dimensiones.

2.1. Ejercicio 1 - Reducción de dimensiones

Para el primer ejercicio, los experimentos consistieron en variar la cantidad de épocas, la regla de aprendizaje y la cantidad de dimensiones de salida. Intentamos variar estos factores y observar diferencias en los gráficos 3D generados en base a las nuevas dimensiones.

Comenzamos realizando clasificaciones con 100 épocas. Al obtener resultados imprecisos subimos la cantidad a 200. Allí observamos una clasificación mejor, más marcada la separación geométrica entre los distintos clusters.

Luego comparamos los resultados obtenidos entre las distintas reglas. Repetimos las ejecuciones variando la regla entre Oja y Sanger para comprobar su eficacia.

Por último, decidimos cambiar la cantidad de dimensiones de sálida, esperando una mejora en las clasificaciones.

2.1.1. Regla de Oja - 200 épocas

En los primeros experimentos, fuimos probando la calidad de la clasificación utilizando solamente 100 épocas. La decisión se basó en un tema de performance de la red. Cada ejecución tenía una cierta demora, por lo que empezamos probando con pocas épocas. Mostramos los primeros resultados que obtuvimos, utilizando la regla de Oja:

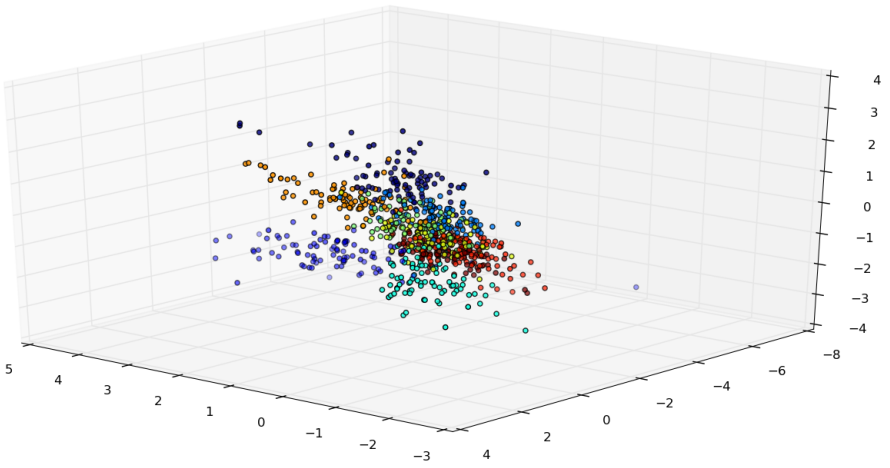

Figura 1: Oja - 3 dimensiones - 200 épocas - Datos Entrenamiento

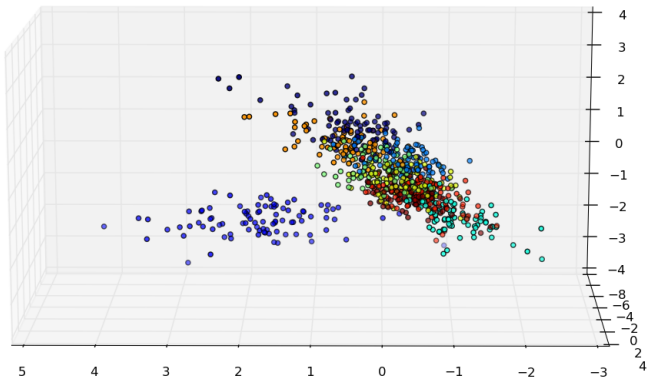

Figura 2: Oja - 3 dimensiones - 200 épocas - Datos Entrenamiento

2.2. Ejercicio 2

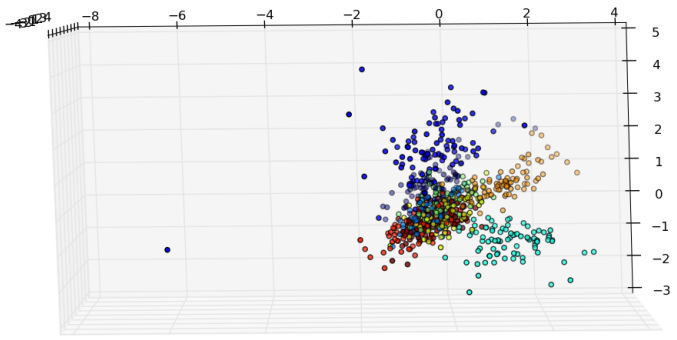

Figura 3: Oja - 3 dimensiones - 200 épocas - Datos Entrenamiento

3. Soluciones Óptimas Propuestas

4. Conclusión

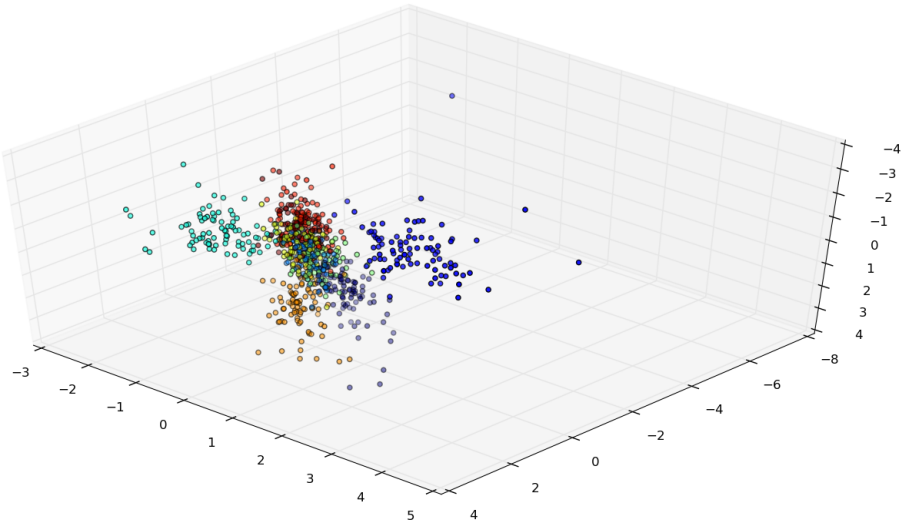

Figura 4: Oja - 3 dimensiones - 200 épocas - Datos Entrenamiento

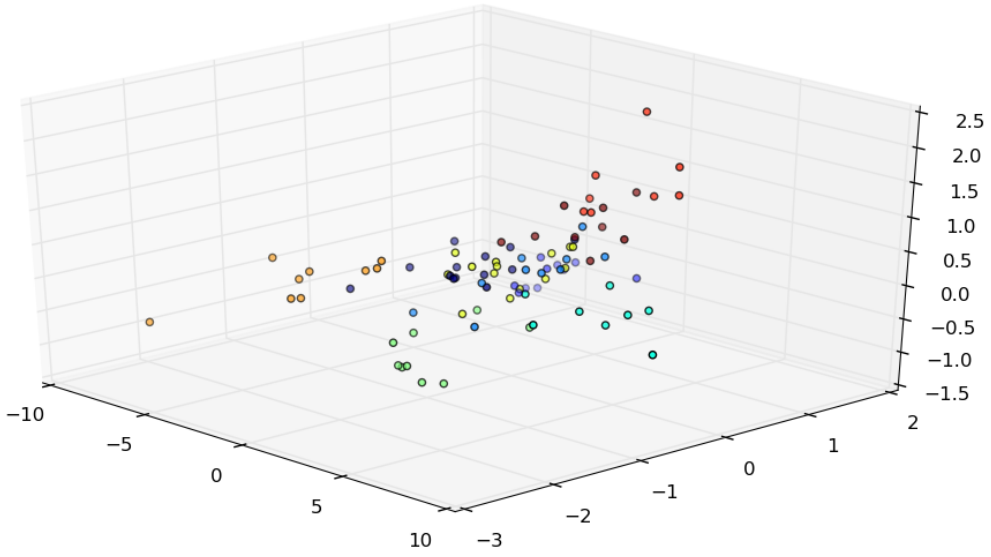

Figura 5: Oja - 3 dimensiones - 200 épocas - Datos Validación

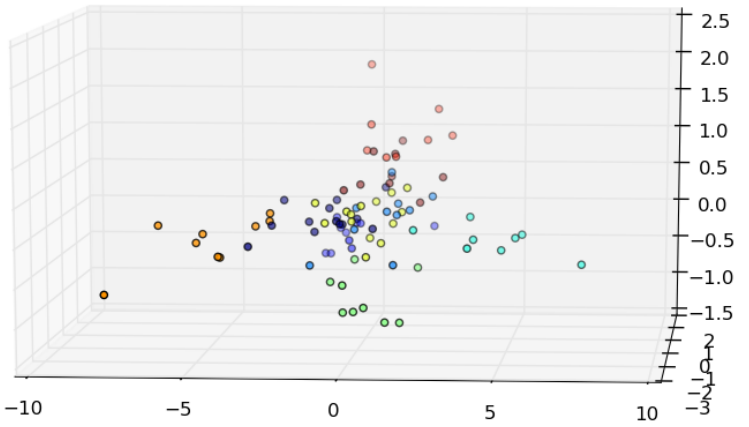

Figura 6: Oja - 3 dimensiones - 200 épocas - Datos Validación

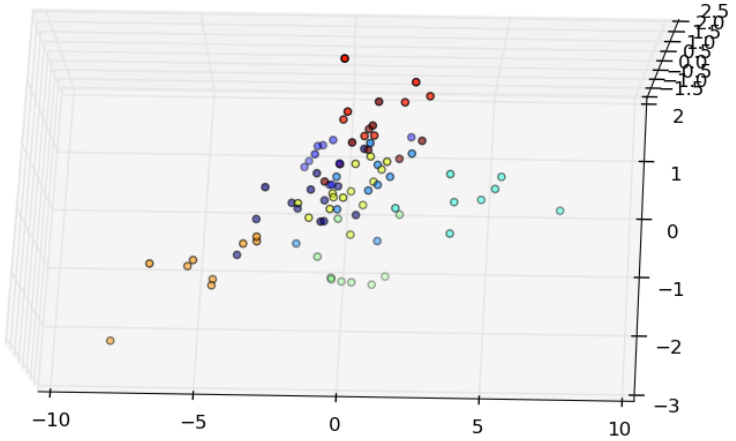

Figura 7: Oja - 3 dimensiones - 200 épocas - Datos Validación