



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Redes Neuronales

Trabajo práctico 2

Resumen

Entrenamiento de Redes Neuronales

Integrante	LU	Correo electrónico
Negri, Franco	893/13	franconegri2004@hotmail.com
?, ?	?/?	?@?.com

Palabras claves:

TP

Índice

1. Introduccion	2
2. Desarrollo	2
2.1. PCA	2
2.1.1. Implementación	2
2.1.2. Experimentación	3
2.2. Mapeo de Características	3
3. Conclusiones	3

1. Introduccion

En este trabajo, nos disponemos a utilizar diferentes tecnicas de aprendizaje no supervisado para clasificar textos con ciertas características. Los mismos consisten en descripciones de diversas compañías y nuestro objetivo será lograr clasificar cada una en una categoria correspondiente. Contamos, ademas, con las verdaderas categorias de cada compañía para realizar una validación de los datos.

2. Desarrollo

2.1. PCA

EL primer modelo utilizado para intentar clasificar los datos será el de Analisis de Componentes Principales. Para ello utilizaremos dos algoritmos basados en aprendizaje Hebbiano y reduciremos las instancias de entrenamiento a 3 dimensiones. Lo que esperamos observar es que aquellas instancias que pertenecen a una misma clase de empresa se encuentran cercas unas de otras, pudiendo observar ñuevde puntos bien definidas.

2.1.1. Implementación

En particular los algoritmos utilizados serán los de *Oja* y *Sanger*. Teniendo una complejidad computacional identica y siendo los algoritmos muy similares, lo distintivo entre estos dos metodos es que *Sanger* ordenará las componentes prinsipales de mayor a menor de acuerdo a sus autovalores mientras que *Oja* no.

El pseodocodigo utilizado para aprendizaje del algoritmo *Oja* será:

- 1: Para toda instancia de entrenamiento, x
- 2: $y = x.W$
- 3: $\tilde{x} = y.W^T$
- 4: $\Delta W = learning_rate((x - \tilde{x})^T.y$

Algorithm 1: Algoritmo De Oja

Mientras que el de *Sanger*

- 1: U = Matriz Triangular Superior Con 1s
- 2: Para toda instancia de entrenamiento, x
- 3: $y = x.W$
- 4: $\tilde{x} = W(y^T.U)$
- 5: $\Delta W = learning_rate((x^T - \tilde{x}).y$

Algorithm 2: Algoritmo De Sanger

Utilizando el paquete numpy de python es posible traducir este codigo de manera casi exacta y de esa manera aprovechar las optimizaciones matriciales que se realizan sobre los datos.

2.1.2. Experimentación

Para la experimentación entrenamos la red con parte del set de datos que nos fue entregado

2.2. Mapeo de Características

En este apartado construiremos un modelo de mapeo de características auto-organizado con la intención de clasificar los documentos en un arreglo de dos dimensiones. Para ello utilizaremos el algoritmo de Kohonen sobre los datos de entrenamiento y una vez que la red haya convergido, graficaremos las las respuestas obtenidas en el plano.

3. Conclusiones