DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TP2

Fecha de entrega: 27/10/2016 | Redes Neuronales Artificiales

Integrante	LU	Correo electrónico
Panarello, Bernabé	194/01	bpanarello@gmail.com
Uhrich, Verónica	021/10	veronicauhrich@gmail.com
Villanueva, Paula	701/10	villanuevapaulasofia@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

Índice

1. Reducción de dimensiones

1.1. Introducción

Disponemos de un dataset de Bag of Words (BOW) que representan descripciones de texto correspondientes a compañías Brasileras clasificadas en nueve categorías distintas. Cada BOW contiene 856 atributos correspondientes a frecuencias de palabras y, debido a la alta dimensionalidad, se busca reducir el conjunto de datos a 3 dimensiones mediante las reglas de aprendizaje de Oja y Sanger.

Regla Oja:

- $\Delta W_{ij} = \eta(x_i - \widetilde{x_i})y_j$
- $\widetilde{x_i} = \sum_{j=1}^{m} y_j.W_{ij}$, donde m es la cantidad de outputs.

Regla Sanger:

- $\Delta W_{ij} = \eta(x_i - \widetilde{x_i})y_j$
- $\widetilde{x_i} = \sum_{k=1}^{j} y_k.W_{ik}$

1.2. Modelo

Nuestro dataset contiene 900 entradas, y cada entrada contiene 856 atributos.

- $X \in \mathbb{R}^{856}$
- $Y \in \mathbb{R}^{3}$
- $W, \Delta W \in \mathbb{R}^{856\times 3}$

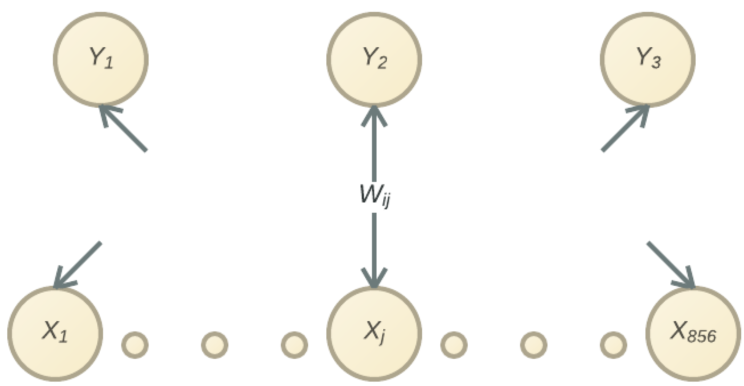

Figura 1: Modelo de red

2. Mapeo de características

2.1. Introducción

Utilizando el mismo dataset de Bag of Words (BOW) del ejercicio anterior debemos construir un modelo de mapeo de características auto-organizado que clasifique automáticamente los documentos en un arreglo de dos dimensiones. Este problema será resuelto utilizando Kohonen.

Kohonen:

- Propone un orden topológico y un modelo competitivo
- Función de transferencia

$$O_i = f(\varepsilon, W_i) donde W_i = (w_{i1}, ..., w_{in}) \in \mathbb{R}_n$$

- PRINCIPIO DE ADAPTACIÓN: Consiste en detectar la unidad cuyos parámetros sean más parecidos a la entrada ε.
  - Establece un orden topológico y solo se cambian los parámetros de la unidad seleccionada y los de sus vecinas.
  - El cambio es en dirección de incrementar la sililaridad entre W_i y ε.
  - La magnitud del cambio debe garantizar estabilidad asintótica.

  Estas tres cosas hacen que la función de densidad de probabilidades de W_i tienda a aproximar la densidad $P(\varepsilon)$.

- MAPAS DE KOHONEN: Es una propuesta de implementación la cual define:
  - Que la unidad seleccionada en tiempo t será c tal que:

  $$||\varepsilon(t) - W_c(t)|| = min\{\varepsilon(t) - W_i(t)\}$$

  - Se establece una topología de entrada a partir de establecer funciones de vecindad, esto fuerza a que los pesos no crezcan de una forma desmedida respecto de los pesos de las unidades vecinas. Esto hace que los entornos sean amplios al principio pero pequeños al final, hasta limitarse solo a la unidad seleccionada.
  - REGLA DE KOHONEN: Actualización de los W_i.

  $$W_i(t+1) = \begin{cases} W_i(t) + \alpha(t)[\varepsilon(t) - W_i(t)] & i \in N_c \\ W_i(t) & i \notin N_c \end{cases}$$

  - $\eta(t)$ es el coeficiente de aprendizaje dinámico, decreciente en el tiempo.

  $$\Delta W_{ij} = \alpha(t)[\varepsilon(t) - W_i(t)]$$

  $$\alpha(t) = \eta\Lambda(i, c)$$

  $$\Lambda(i, c) = \begin{cases} 1 & i = c \\ decreceamayordistanciaentreiyc \end{cases}$$

- TEOREMA DE KOHONEN: Con probabilidad 1 los W_i se ordenan de forma ascendientes o descendientes cuando $t \to \infty$ si $\alpha(t) \to 0$ con suficiente lentitud.

2.2. Modelo

Al igual que en el ejercicio anterior nuestro dataset tiene 900 entradas, y cada una de ellas contiene 856 atributos.

Para que un modelo sea auto-organizado debemos elegir el sigma σ y el eta η adecuados. El sigma será utilizado como parámetro de la Gauseana y medirá la dispersión, cuantas neuronas vecinas a la neurona activada modificarán sus pesos. Por otro lado, el η será el coeficiente de aprendizaje de nuestra red.

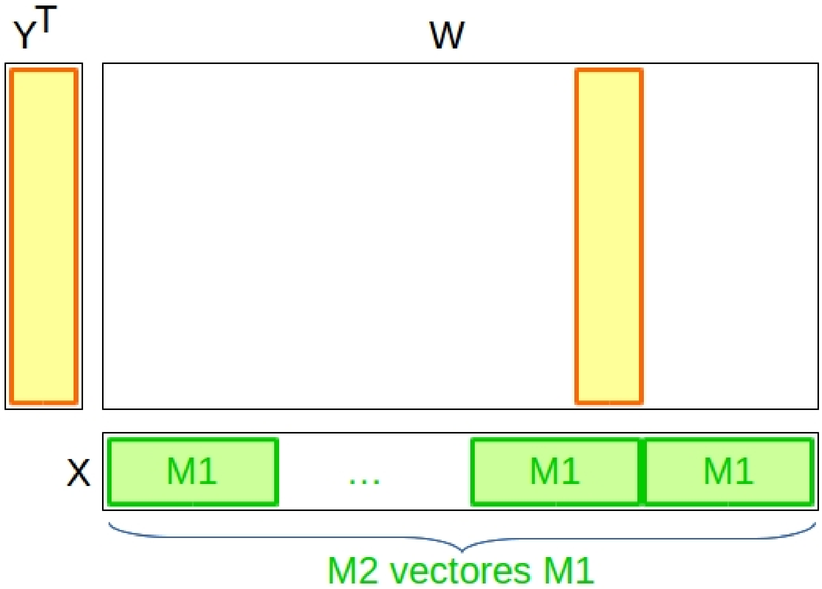

Figura 2: Comparación

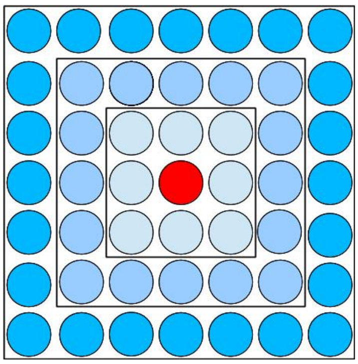

Figura 3: Función de vecindad.

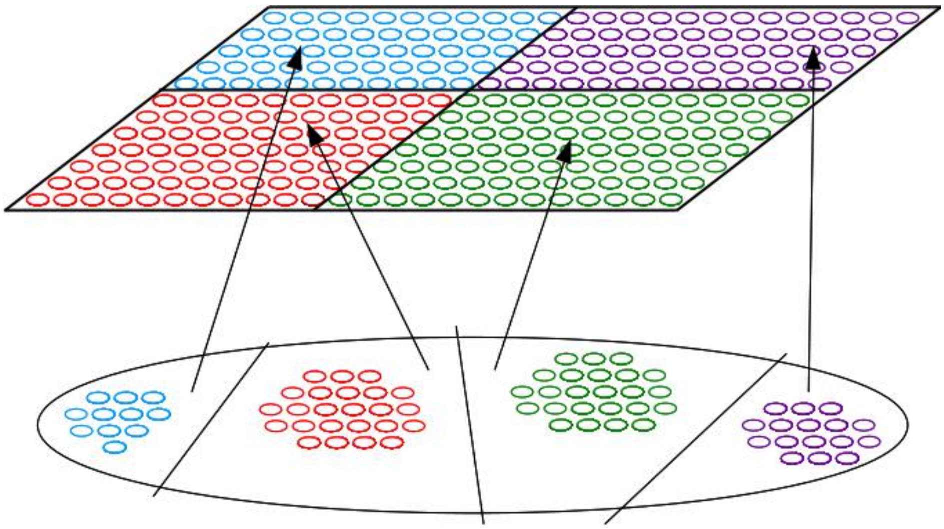

Figura 4: Ejemplo con cuatro clases.