Redes Neuronales

Redes Neuronales competitivas

carlos.andres.delgado@correounivalle.edu.co

Carlos Andrés Delgado S.

Facultad de Ingeniería. Universidad del Valle

Marzo de 2019

Contenido

Contenido

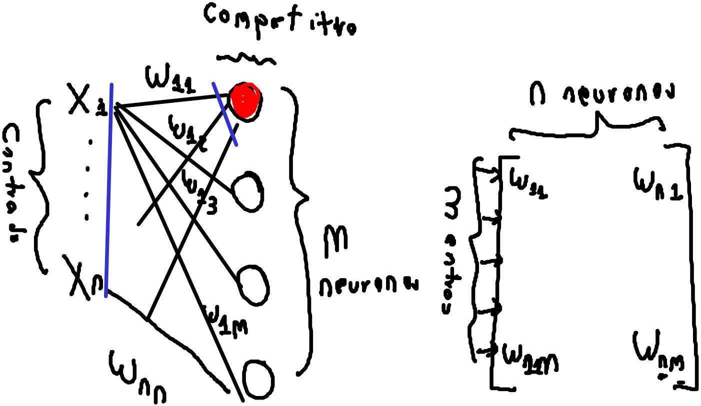
Introducción

Definiciones

- Existe una capa llamada capa competitiva
- la capa competitiva se conecta totalmente con los nodos de entrada
- Se utilizan conexiones inhibitorias, las cuales tienen dos estados: encendido y apagado.
- El potencial sináptico de la neurona i para una entrada $[x_1, x_2, ...x_n]$ está dado por la expresión:

$$h_i = w_{i1}x_1 + w_{i2}x_2 + ... + w_{iN}x_n - \Theta_i$$

Donde $\Theta_i = \frac{1}{2}(w_{i1}^2 + w_{i2}^2 + ... + w_{iN}^2)$



X1
Xn
W11
W12
W13
W1m
Wnn
m
neuronas
n neuronas
W11
Wn1
W1m
Wnm

Introducción

Definiciones

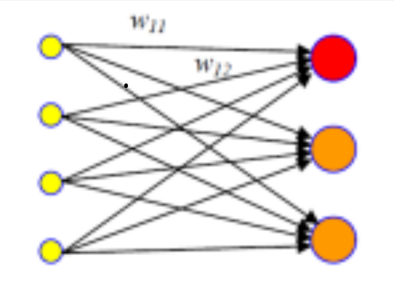
- Se utilizan en problemas de clasificación con una clase (Es o no es de una clase)
- Tenemos aprendizaje supervisado y no supervisado

Contenido

Aprendizaje no supervisado

Definiciones

- Una red competitiva es constituida por N señales de entrada y M unidades de proceso (neuronas artificiales)
- Para cada patrón de entrada sólo se activa una neurona (Aquella que es ganadora)



Aprendizaje no supervisado

Definiciones

- El estado de la unidad de proceso i es una variable booleana y_i
- La variable y_i viene dada por la expresión:
  $$y_i = \begin{cases} 1 & \text{si} \quad h_i = \max_k\{h_1, h, 2, ..., h_M\} \\ 0 & \text{si} \quad \text{otro caso} \end{cases}$$
- Cada entrada a la red es un vector $[x_1, x_2, ...x_n]$ y para el cual sólo se activa una neurona, permaneciendo las restantes desactivadas

Universidad del Valle

Aprendizaje no supervisado

¿Como se determinan los pesos?

- Se utiliza un proceso de aprendizaje no supervisado
- Se busca que se active sólo la neurona cuya vector de pesos sinápticos sea el más similar a la entrada
- Por esto, los pesos sinápticos son la mejor representación del conjunto de patrones
- Se busca demostrar que la unidad ganadora es aquella cuyo valor de pesos sinápticos es que más se parece al vector de entrada

Universidad del Valle

Aprendizaje no supervisado

¿Como se determinan los pesos?

- Se utiliza la distancia euclidiana entre los pesos y la entrada:

$$d(x, w_i) = \sqrt{(x_1 - w_{i1})^2 + ... + (x_N - w_{iM})^2}$$

Aprendizaje no supervisado

Teorema

- Si r es la neurona ganadora entonces cumple:

$$d(x, w_r) \leq d(x, w_k), 1 \leq k \leq M$$

- Ahora vamos a introducir el concepto de error cuadrático:

$$\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{P} a_{ij}(d(x(j), w_i))^2$$

Donde: $a_{ij} = \begin{cases} 1 & \text{si} \quad x(j)\text{Es de clase C} \\ 0 & \text{si} \quad \text{otro caso} \end{cases}$

Teorema

- En este punto se introduce la regla de aprendizaje
- Para cada iteración k se busca determinar los nuevos valores en los vectores de aprendizaje por regla del gradiente descendiente:

$$\nabla w_i(k) = -\epsilon \frac{\partial E}{\partial w_i(k)}$$

- Realizando el álgebra del caso se llega a que la regla de aprendizaje es: $\nabla w_i(k) = \begin{cases} \underline{\epsilon(k)}\underbrace{(x(k) - w_i(k))}_{\text{Error}} & \text{si} \quad i = r \\ 0 & \text{si} \quad i \neq r \end{cases}$

Aprendizaje no supervisado

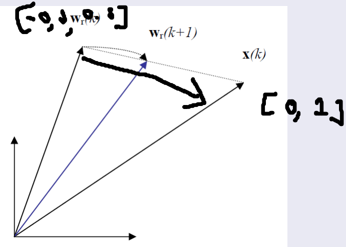
Teorema

- Para realizar el proceso de aprendizaje se van a determinar los vectores de pesos sinápticos
- Se realiza un proceso iterativo, el cual busca minimizar el error cuadrático, mediante el método de gradiente descendiente
- La regla de aprendizaje puede ser de dos formas:
    - Actualizar los pesos cada vez que se introduce un patrón de entrada, el cual es aprendizaje individualizado
    - Actualizar los pesos después de introducir todos los patrones, que es el conocido como aprendizaje por lotes

Aprendizaje no supervisado

Teorema

- Los pesos sinápticos solo se actualizan en las neurona ganadora



- Se introduce un factor de aprendizaje ϵ_0 el cual se modifica en cada iteración k, así $\epsilon(k) = \epsilon_0(1 - \frac{k}{T})$ Donde T es el número total de iteraciones hasta concluir el aprendizaje.

Aprendizaje individualizado

Procedimiento

1. Elegir como vectores de pesos sinápticos iniciales P patrones de entrenamiento aleatorios y poner $k = 1$. Se tienen M neuronas.
2. Elegir un patrón de entrenamiento
3. Calcular los potenciales iniciales $h_1(k), h_2(k), ..., h_n(k)$
4. Escoger el mayor potencial sináptico $h_r(k) = (max)(h_1(k), h_2(k), ..., h_n(k))$
5. Actualizar w_r (r es la neurona ganadora) así:

$$w_r(k+1) = w_r(k) + \epsilon(k)(x(k) - w(k))$$

Universidad del Valle

Aprendizaje individualizado

Procedimiento

6. Se actualiza la taza de aprendizaje $\epsilon(k) = \epsilon_0(1 - \frac{k}{T})$
7. Si $k = T$ parar, en otro caso $k = k + 1$ e ir a paso 1. El valor T puede ser especificado o usando un valor de aprendizaje pequeño.

Universidad del Valle

Aprendizaje por lotes

Procedimiento

1. Elegir como vectores de pesos sinápticos iniciales P patrones de entrenamiento aleatorios y poner $k = 1$. Se tienen M neuronas.
2. Calcular los potenciales sinápticos para cada patrón de entrenamiento $h_1(k), h_2(k), ..., h_n(k)$
3. Determinar la neurona ganadora r $h_r(k) = (max)(h_1(k), h_2(k), ..., h_n(k))$ para cada patrón de entrada. Poner
$$a_y = \begin{cases} 0 & \text{si} \quad i \text{es la unidad ganadora para x(j)} \\ 1 & \text{si} \quad \text{otro caso} \end{cases}$$

Aprendizaje por lotes

Procedimiento

4. Actualizar cada w_i así:

$$w_i(k+1) = w_i(k) + \epsilon(k) \boxed{\sum_{j=1}^{p} a_y(x(k) - w(k))}$$

→1 1*
2 2'
→3 1'
→4 2*

5. Se actualiza la taza de aprendizaje $\epsilon(k) = \epsilon_0(1 - \frac{k}{T})$
6. Si $k = T$ parar, en otro caso $k = k + 1$ e ir a paso 1. El valor T puede ser especificado o usando un valor de aprendizaje pequeño.

Universidad del Valle

Contenido

Universidad del Valle

Aprendizaje supervisado

Conceptos

- Una red neuronal competitiva está compuesta por N entradas y M unidades de proceso
- La dinámica de computación esta dada por:

$$y_i = \begin{cases} 1 & \text{si} \quad h_i = \max_k\{h_1, h, 2, ..., h_M\} \\ 0 & \text{si} \quad \text{otro caso} \end{cases}$$

- Para cada entrada se selecciona la neurona que tenga mayor potencial sináptico, el cual está dado por:

$$h_i = w_{i1}x_1 + w_{i2}x_2 + ... + w_{iN}x_n - \Theta_i$$

Donde $\Theta_i = \frac{1}{2}(w_{i1}^2 + w_{i2}^2 + ... + w_{iN}^2)$

Aprendizaje supervisado

¿Como se determinan los pesos sinápticos?

En aprendizaje supervisado podemos decir si un patrón corresponde o no a una clase.

- Si la unidad de aprendizaje es de la clase correcta acercamos el vector de pesos de la unidad ganadora a la entrada
- Si es de la clase incorrecta, alejamos el peso sináptico de la unidad ganadora al vector de entrada

Por lo tanto la regla de aprendizaje es:

$$\Delta w_r(k) = \begin{cases} \epsilon_r(x(k) - w_r(k)) & \text{si} \quad r = s \\ -\epsilon_r(x(k) - w_r(k)) & \text{si} \quad r \neq s \\ 0 & \text{si} \quad i \neq r \end{cases}$$

la entrada x es de la clase s y la unidad ganadora es r

Aprendizaje supervisado

¿Como se determinan los pesos sinápticos?

Para mejorar la convergencia, cada unidad tiene su factor de aprendizaje y se comporta de acuerdo a la siguiente regla:

$$\epsilon_r(k+1) = \begin{cases} \frac{\epsilon_r(k)}{1+\epsilon_r(k)} & \text{si} \quad r = s \\ \frac{\epsilon_r(k)}{1-\epsilon_r(k)} & \text{si} \quad r \neq s \end{cases}$$

La tasa de aprendizaje se disminuye si su predicción es correcta y se aumenta en caso contrario.

Aprendizaje supervisado

Procedimiento

1. Elegir k_i prototipos iniciales para la clase i, $i = 1, 2, ..., m$.Estos serán los vectores sinápticos de las unidades de proceso.
2. (Késima iteración) Seleccionar aleatoriamente un patrón del conjunto de entrenamiento $x(k)$
3. Escoger el mayor potencial sináptico $h_r(k) = (max)(h_1(k), h_2(k), ..., h_n(k))$

Universidad del Valle

Aprendizaje individualizado

Procedimiento

5. Fase de aprendizaje:
   - Si r es la unidad ganadora y corresponde a la misma clase que la entrada x(k) entonces se modifica de la siguiente forma:

$$w_r(k+1) = w_r(k) + \epsilon(k)(x(k) - w(k))$$

   - Si no es de la misma clase

$$w_r(k+1) = w_r(k) - \epsilon(k)(x(k) - w(k))$$

Aprendizaje individualizado

Procedimiento

6. Repetir el paso 2 modificando el factor de aprendizaje de la unidad ganadora:

$$\epsilon_r(k+1) = \begin{cases} \frac{\epsilon_r(k)}{1+\epsilon_r(k)} & \text{si} \quad r = s \\ \frac{\epsilon_r(k)}{1-\epsilon_r(k)} & \text{si} \quad r \neq s \end{cases}$$

Universidad del Valle

Referencias I

Du, K.-L. and Swamy, M. N. S. (2010).
Neural Networks in a Softcomputing Framework.
Springer Publishing Company, Incorporated, 1st edition.

Gómez, A. M. (2014).
Redes neuronales artificiales: The Self-Organizing Maps (SOM) para el reconocimiento de patrones.
I(1):27–38.

Heaton, J. (2008).
Introduction to Neural Networks with Java.
Heaton Research.

Referencias II

Perez Muñoz, J. (2017).
Curso de modelos computacionales.
http://www.lcc.uma.es/~munozp/.
Accessed: Ocubre-2017.

¿Preguntas?

Próximo tema:
Redes de base radial