

Redes Neuronales

Redes Neuronales de base radial
carlos.andres.delgado@correounivalle.edu.co

Carlos Andrés Delgado S.

Facultad de Ingeniería. Universidad del Valle

Marzo de 2019



Contenido


- 1 Funciones de base radial
- 2 Modelo neuronal
- 3 Algoritmo Kmeans
- 4 Entrenamiento
- 5 Redes de base radial frente a perceptrón multicapa

Contenido

- 1 Funciones de base radial
- 2 Modelo neuronal
- 3 Algoritmo Kmeans
- 4 Entrenamiento
- 5 Redes de base radial frente a perceptrón multicapa

Funciones de base radial

Definiciones

- Son funciones cuya salida depende de la distancia a un punto denominado centro
- Son funciones simétricas con respecto a 0 
- Se definen al menos dos parámetros:
 - Centro: Punto donde la función posee un máximo
 - Anchura: Magnitud de la variación de la función según la distancia del centro.

Funciones de base radial

Definiciones

$$e^{-\frac{1}{2}}$$

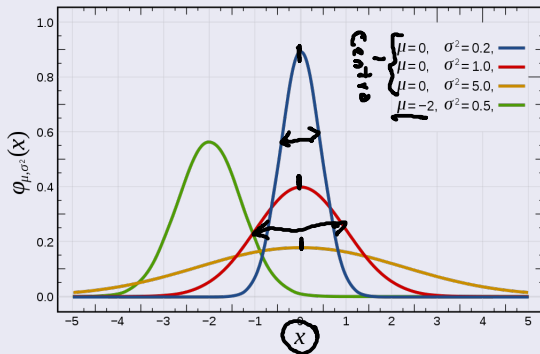


Figura: Función de Gauss. Tomado de Wikipedia.

Contenido

- 1 Funciones de base radial
- 2 Modelo neuronal**
- 3 Algoritmo Kmeans
- 4 Entrenamiento
- 5 Redes de base radial frente a perceptrón multicapa

Definiciones

- Se tiene un modelo similar al perceptrón multicapa
- La capa de entrada recibe señales y no realiza ningún procesamiento
- Capa oculta: Realiza una transformación local de las entradas y no lineal de esas señales, a diferencia de perceptrón
- Capa de salida: Realiza una combinación lineal de las señales y actúa como salida

Modelo neuronal

Definiciones

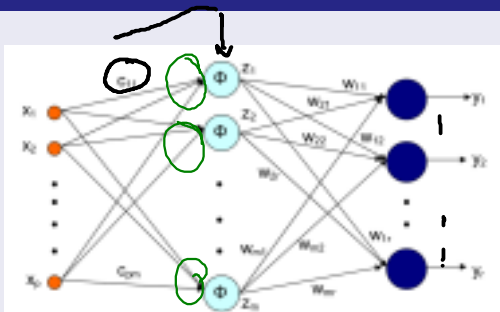


Figura: Arquitectura red neuronal base radial [Pinedo, 2015].

Modelo neuronal

Capa de salida

- Cada elemento procesado calculado su valor neto como una combinación lineal de las salidas de la capa oculta
- Para un patrón dado $[x_1, x_2, \dots, x_n]$ la salida cada elemento k de la capa de salida se obtiene así:

$$y_k(n) = \sum_{i=1}^m \underline{w_{ik}} \underline{z_i(n)} + u_k, k \in [1, 2, \dots, r] \quad (1)$$

Donde w_{ik} son los pesos asociados a cada elemento k de la salida y el elemento i de la capa oculta y $z_i(n)$ son las salidas procesadas obtenidas de la capa oculta. En total tenemos m neuronas en la capa oculta y r en la capa de salida. u_k es el umbral de cada neurona de salida.

Modelo neuronal

Capa oculta

Cada elemento de procesamiento tiene asociada una función de base radial de tal manera que representa una clase o categoría, donde dicha clase viene dada por $(\underline{C_i}, d_i)$. C_i representa el centro del cluster (pesos asociados a cada neurona i) y d_i representa la desviación, anchura o dilatación de la función de base radial.

Modelo neuronal

Capa oculta

La salida de cada elemento de la capa oculta $z(n)$ se calcula como la distancia que existe entre el patrón $X(n)$ al centro del cluster C_i ponderada inversamente por d_i , de la siguiente forma:

$$z_i(n) = \phi \left(\frac{\sqrt{\sum_{j=1}^p (x_j(n) - c_{ij})^2}}{d_i} \right) \quad (2)$$

(Handwritten green circle around the fraction and a green line pointing to the exponent 2 in the denominator of the fraction)

Donde ϕ es una función de base radial. La más común es

$$\phi(r) = e^{-\frac{r^2}{2}}.$$

$$[x_1, x_2 \dots x_p] \quad [c_1, c_2 \dots c_p]$$
$$(x_1 - c_1)^2 + (x_2 - c_2)^2 \dots$$



Características

- Las funciones de base radial tienen un carácter local, ya sólo se basan en el patrón de entrada actual y no en el total
- Estas funciones alcanzan un máximo cuando el patrón está próximo a la neurona (centro)
- A medida que se aleja el patrón del centro, la salida de la función disminuye
- Las salidas de las redes neuronales de base radial son una combinación lineal de gaussianas

Contenido

- 1 Funciones de base radial
- 2 Modelo neuronal
- 3 Algoritmo Kmeans**
- 4 Entrenamiento
- 5 Redes de base radial frente a perceptrón multicapa

Algoritmo Kmeans

Introducción

- Es un algoritmo de agrupamiento
- Permite generar la partición de un conjunto de n datos en k grupos
- Es un método ampliamente utilizado en minería de datos
- En RBF lo requerimos para calcular los centros en la técnica de aprendizaje híbrido

Algoritmo Kmeans

Algoritmo

- 1 Es un proceso iterativo, en que cada iteración determinamos los centros
- 2 Se asume un espacio n dimensional de características de entrada
- 3 Inicialmente, calculamos los k centros aleatoriamente
 $c_i = (x_1, x_2, \dots, x_n)$
- 4 Tomamos las distancias a cada uno de los puntos de entrada y asignamos al centro más cercano

Algoritmo Kmeans

Algoritmo

- 5 Calculamos para cada cluster asignado su centro
- 6 Si los centros son distintos a los iniciales, volvamos a realizar el proceso
- 7 El algoritmo se detiene cuando de una iteración a otra no hay cambio en los centros

Algoritmo Kmeans

Ejemplo

Dados los siguiente datos:

x_1	x_2	x_3
1	4	3
2	1	7
3	1	4
7	2	1

Tomamos $k = 2$

Algoritmo Kmeans

Ejemplo

Calculamos dos centros aleatorios $C_1 = (1, 1, 1)$, $C_2 = (4, 4, 4)$. Para la distancia a los centros se puede usar distancia geométrica o manhattan. En nuestro caso manhattan.

x_1	x_2	x_3	$\text{dis}(C_1)$	$\text{dis}(C_2)$	asig
1	4	3	5	4	C_2
2	1	7	7	8	C_1
3	1	4	5	4	C_2
7	2	1	7	8	C_1

Los nuevos centros son: $C_1 = (4, 5, 1, 5, 4)$, $C_2 = (2, 5, 3, 5, 5)$

Algoritmo Kmeans

Ejemplo

Centros son: $C_1 = (4,5,1,5,4)$, $C_2 = (2,5,3,5,5)$

x_1	x_2	x_3	$\text{dis}(C_1)$	$\text{dis}(C_2)$	asig
1	4	3	7	4	C_2
2	1	7	6	5	C_2
3	1	4	2	4	C_1
7	2	1	6	9	C_1

Los nuevos centros son: $C_1 = (5,1,5,2,5)$, $C_2 = (1,5,2,5,5)$

Algoritmo Kmeans

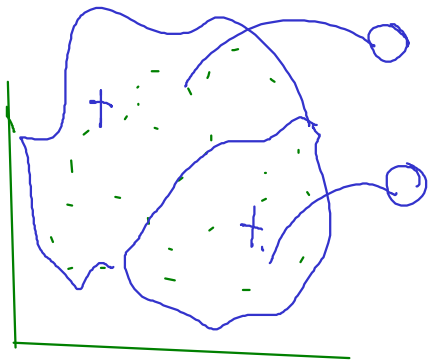
Ejemplo

Centros son: $C_1 = (5, 1, 5, 2, 5)$, $C_2 = (1, 5, 2, 5, 5)$

x_1	x_2	x_3	$\text{dis}(C_1)$	$\text{dis}(C_2)$	asig
1	4	3	7	4	C_2
2	1	7	8	4	C_2
3	1	4	4	4	C_1^*
7	2	1	4	9	C_1

* Lo seleccionamos por conveniencia.

Los nuevos centros son: $C_1 = (5, 1, 5, 2, 5)$, $C_2 = (1, 5, 2, 5, 5)$.
Hemos terminado ya que los centros no han cambiado.



Contenido

- 1 Funciones de base radial
- 2 Modelo neuronal
- 3 Algoritmo Kmeans
- 4 Entrenamiento**
- 5 Redes de base radial frente a perceptrón multicapa

Entrenamiento

El entrenamiento de este tipo de redes determina todos los parámetros de la red

- Parámetros de salida: Pesos. Los cuales se determinan por la optimización del espacio de entradas
- Parámetros de capa oculta: Centros y desviaciones. Estos se determinan en base a las salidas deseadas

Para este tipo de redes se tienen varias estrategias de entrenamiento, la más popular es el entrenamiento híbrido, debido a su facilidad de implementación.

Entrenamiento híbrido

Este consiste en dos pasos:

- Fase no supervisada: Determinación de parámetros en capa oculta
- Fase supervisada: Determinación de pesos en la capa de salida

Entrenamiento híbrido: Fase no supervisada

- Determinación de centros: Algoritmo de K-means o Mapas de Kohonen
- Determinación de desviaciones: Se deben calcular de tal manera que cada neurona de la capa oculta se active de tal forma la solapación de espacios sea lo más pequeña posible.

Entrenamiento híbrido: Fase no supervisada

Existen varias estrategias para la determinación de desviaciones:

- Media uniforme de las distancias euclidianas del centro C_i a los p centros más cercanos

$$d_i = \frac{1}{2} \sum_p \|C_i - C_p\| \quad (3)$$

- Media geométrica de la distancia del centro a sus dos vecinos más cercanos:

$$d_i = \sqrt{\|C_i - C_t\| * \|C_i - C_s\|} \quad (4)$$

Donde C_t y C_s son los más cercanos a C_i

Entrenamiento híbrido: Fase supervisada

- Se utiliza la técnica de corrección del error que vimos en Adeline o MLP
- Minimización del error cuadrático medio
- Las reglas de entrenamiento son:

$$\begin{aligned}w_{ik}(t+1) &= w_{ik}(t) + \epsilon(t^k - y^k)\phi(n) \\u_k(t+1) &= u_k(t) + \epsilon(t^k - y^k)\end{aligned}$$

Donde ϵ es el factor de entrenamiento, u_k el centro, t^k es la salida deseada, w_{ik} es el peso y y^k es la salida obtenida.

Contenido

- 1 Funciones de base radial
- 2 Modelo neuronal
- 3 Algoritmo Kmeans
- 4 Entrenamiento
- 5 Redes de base radial frente a perceptrón multicapa

RBF vs MLP

Comparación

MLP	RBF
Uso de funciones de transferencia sigmoidales	Cada neurona en la capa oculta se especializa en una región del espacio de entradas, es decir en una clase.
Aprendizaje lento: Cambio de pocos pesos por patrón	Aprendizaje rápido: el cambio de pesos sólo afecta a la neurona asociada a dicho peso o a los patrones de la misma clase

RBF vs MLP

Comparación

MLP	RBF
Es robusta y tolerante al ruido. No se requiere un gran número de neuronas con relación al espacio de entradas.	Es necesario en algunos casos un gran número de neuronas en la capa oculta ya que se puede perder generalización. El número de neuronas en capa oculta crece exponencialmente de acuerdo al espacio de entradas.

Referencias I



MacQueen, J. (1967).

Some methods for classification and analysis of multivariate observations.

In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pages 281–297, Berkeley, Calif. University of California Press.



Pinedo, M. (2015).

Curso de sistemas conexionistas.

<http://www.varpa.org/~mgpenedo/>.

Accessed: Marzo-2017.

¿Preguntas?

Próximo tema:
Tendencias en redes neuronales.