

2 on 1[a4paper,border shrink=5mm]

Redes neuronales con funciones de base radial

Diego Milone y Leonardo Rufiner
Inteligencia Computacional
Departamento de Informática

FICH-UNL

Organización: RBF-NN

- Motivación y orígenes RBF
- Arquitectura y modelo matemático RBF-NN
- Método de entrenamiento
 - Adaptación de las RBF: batch k-means
 - Adaptación de las RBF: online k-means
 - Adaptación de los pesos
- Comparación con el perceptrón multicapa
- Anotaciones sobre gaussianas N -dimensionales

RBF-NN: introducción

- Motivación: regiones de decisión

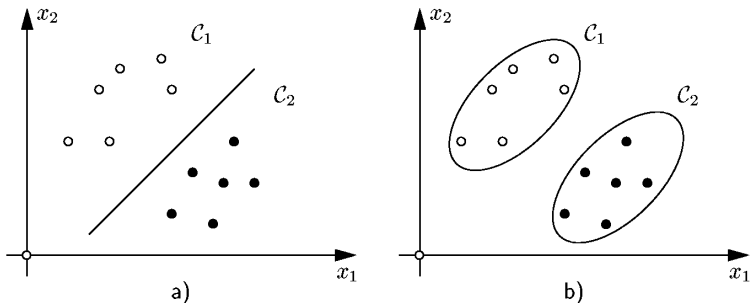


Figura: Espacio de Patrones bidimensionales: a) El perceptrón representa límites de decisión (lineales), b) Las RBF-NN representan clases (a través de las funciones base).

RBF-NN: introducción

- Ejemplo: datos reales vocales

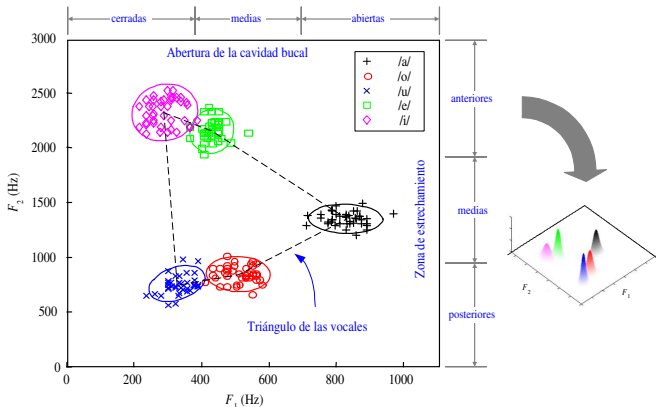


Figura: Mapa de las formantes obtenido a partir de datos experimentales para las vocales del español pronunciadas en forma sostenida por un conjunto de hablantes masculinos.

RBF-NN: introducción

- Orígenes de las RBF: aproximación de funciones

$$f : \mathbb{R}^N \rightarrow \mathbb{R}$$

$$d = f(\mathbf{x})$$

Aproximación:

$$h(\mathbf{x}) = \sum_j w_j \phi(\|\mathbf{x} - \mu_j\|)$$

frecuentemente se utiliza como función de base radial:

$$\phi(\kappa) = e^{-\frac{\kappa^2}{2\sigma^2}}$$

RBF-NN: introducción

- Aproximación:

$$\rho(\mathbf{x}) = \sum_j w_j \phi(\|\mathbf{x} - \mu_j\|)$$

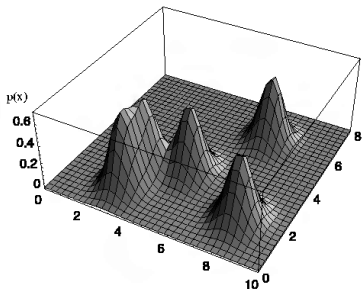


Figura: Ejemplo de aproximación de una función bidimensional $\rho(\mathbf{x})$ mediante 5 gaussianas $\phi(\kappa) = e^{-\frac{\kappa^2}{2\sigma^2}}$.

RBF-NN: generalidades

- Arquitectura

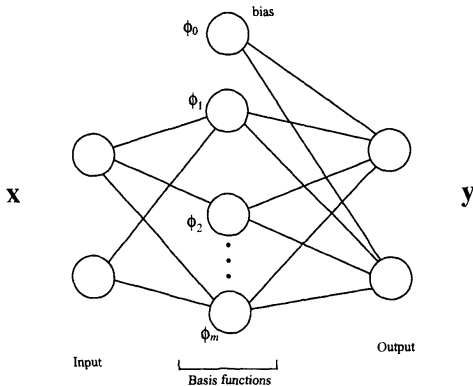


Figura: Arquitectura de una RBF-NN.

RBF-NN: generalidades

- Modelo matemático

$$y_k(\mathbf{x}_\ell) = \sum_{j=1}^M w_{kj} \phi_j(\mathbf{x}_\ell)$$

donde:

$$\phi_j(\mathbf{x}_\ell) = e^{-\frac{\|\mathbf{x}_\ell - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}}$$

- ¿Cuáles son los parámetros a entrenar?

RBF-NN: entrenamiento

- Método 1:
 - Adaptación no supervisada de las RBF
 - Utilizando el método k-means
 - Utilizando mapas autoorganizativos
 - Otros...
 - Adaptación supervisada de los w_{kj} (LMS)
- Método 2:
 - Inicialización por el Método 1
 - Adaptación supervisada de las RBF $\left(\frac{\partial \xi}{\partial \mu_{ji}}, \frac{\partial \xi}{\partial \sigma_j} \right)$
- En general se adaptan RBF y w_{kj} por separado.

Adaptación de las RBF: *k-means*

Método NO-supervisado!

Objetivos:

- Encontrar k conjuntos C_j de forma que:
 - Cada conjunto C_j sea lo más diferente posible de los demás
 - Los patrones \mathbf{x}_ℓ dentro de cada C_j sean lo más parecidos posible entre ellos
- Encontrar el centroide μ_j de cada conjunto C_j

Ecuación de optimización: $\min \left\{ J = \sum_{j=1}^k \sum_{\ell \in C_j} \|\mathbf{x}_\ell - \mu_j\|^2 \right\}$

Adaptación de las RBF: batch k -means

1. Inicialización: se forman los k conjuntos $C_j(0)$ con patrones \mathbf{x}_ℓ elegidos al aleatoriamente.
2. Se calculan los centroides:

$$\boldsymbol{\mu}_j(n) = \frac{1}{|C_j(n)|} \sum_{\ell \in C_j(n)} \mathbf{x}_\ell$$

3. Se reasignan los \mathbf{x}_ℓ al C_j más cercano:

$$\ell \in C_j(n) \Leftrightarrow \|\mathbf{x}_\ell - \boldsymbol{\mu}_j\|^2 < \|\mathbf{x}_\ell - \boldsymbol{\mu}_i\|^2 \quad \forall i \neq j$$

4. Volver a 2 hasta que no se realicen reasignaciones.

Adaptación de las RBF: online k -means

Optimización por método de gradiente:

$$\nabla_{\mu} J = \nabla_{\mu} \left\{ \sum_{j=1}^k \sum_{\ell \in \mathcal{C}_j} \|\mathbf{x}_{\ell} - \mu_j\|^2 \right\} = 0$$

$$\mu_j(n+1) = \mu_j(n) + \eta(\mathbf{x}_{\ell} - \mu_j(n))$$

Adaptación de las RBF: online k -means

1. Inicialización: se eligen k patrones aleatoriamente y se usan como centroides iniciales $\mu_j(0) = \mathbf{x}'_\ell$.

2. Selección:

$$j^* = \arg \min_j \{ \|\mathbf{x}_\ell - \mu_j(n)\| \}$$

3. Adaptación:

$$\mu_{j^*}(n+1) = \mu_{j^*}(n) + \eta(\mathbf{x}_\ell - \mu_{j^*}(n))$$

4. Volver a 2 hasta no encontrar mejoras significativas en J .

Adaptación de los pesos: generalidades

- Al entrenar los pesos, las RBF quedan fijas (ver Figura 4)
- Al estar las RBF fijas se pueden obtener las salidas intermedias para cada patrón de entrada: $\phi(\mathbf{x}_\ell)$
- Con esas salidas intermedias se puede entrenar cada perceptrón simple:

$$\mathbf{y} = \mathbf{W}\phi(\mathbf{x}_\ell)$$

- Métodos de entrenamiento:
 - pseudo-inversa del vector $\phi(\mathbf{x}_\ell)$
 - gradiente descendiente sobre el error cuadrático instantáneo (LMS)

Adaptación de los pesos: método LMS

$$e_k(n) = y_k(n) - d_k(n)$$

$$\xi(n) = \frac{1}{2} \sum_k e_k^2(n) = \frac{1}{2} \sum_k \left(\sum_j w_{kj}(n) \phi_j(n) - d_k(n) \right)^2$$

$$\frac{\partial \xi(n)}{\partial w_{kj}(n)} = (y_k(n) - d_k(n)) \frac{\partial}{\partial w_{kj}} \left(\sum_j w_{kj}(n) \phi_j(n) - d_k(n) \right)$$

$$\frac{\partial \xi(n)}{\partial w_{kj}(n)} = e_k(n) \phi_j(n)$$

Adaptación de los pesos: método LMS

Regla de aprendizaje:

$$w_{kj}(n+1) = w_{kj}(n) - \eta e_k(n) \phi_j(n)$$

$$w_{kj}(n+1) = w_{kj}(n) - \eta \left(\sum_i w_{ki}(n) \phi_i(n) - d_k(n) \right) \phi_j(n)$$

Comparación RBF-NN vs. MLP

RBF-NN

1 capa oculta

distancia a prototipos gaussianos

representaciones locales sumadas

convergencia más simple (linealidad)

entrenamiento más rápido

arquitectura más simple

combinación de diferentes paradigmas de aprendizaje

MLP

p capas ocultas

hiperplanos sigmoideos

representaciones distribuidas combinadas

Anotaciones sobre gaussianas N -dimensionales

Concepto, interpretación gráfica y forma matricial

- Forma general $\rightarrow \mathbf{x}, \boldsymbol{\mu}_j \in \mathbb{R}^N, \mathbf{U}_j \in \mathbb{R}^{N \times N}$:

$$\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_j, \mathbf{U}_j) = \frac{1}{(2\pi)^N |\mathbf{U}_j|^{1/2}} \cdot e^{-\frac{1}{2} [(\mathbf{x} - \boldsymbol{\mu}_j)^T \mathbf{U}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j)]}$$

- Caso simplificado 1 $\rightarrow \mathbf{U}_j \in \mathbb{R}^{N \times N}$, diagonal general:

$$\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_j, \mathbf{U}_j) = \frac{1}{(2\pi)^N \sqrt{\sum_{k=1}^N \sigma_{jk}^2}} \cdot e^{-\frac{1}{2} \sum_{k=1}^N \frac{(x_k - \mu_{jk})^2}{\sigma_{jk}^2}}$$

Anotaciones sobre gaussianas N -dimensionales

Concepto, interpretación gráfica y forma matricial

- Caso simplificado 2 $\rightarrow \mathbf{U}_j \in \mathbb{R}^{N \times N}$, diagonal general:

$$\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_j, \mathbf{U}_j) = \frac{1}{(2\pi)^N \sqrt{N\sigma}} \cdot e^{-\frac{1}{2\sigma^2} \sum_{k=1}^N (x_k - \mu_{jk})^2}$$

- Caso simplificado 3 $\rightarrow \mathbf{U}_j = \mathbf{I}$:

$$\mathcal{N}'(\mathbf{x}, \boldsymbol{\mu}_j) = e^{-\frac{1}{2} \sum_{k=1}^N (x_k - \mu_{jk})^2}$$

(las varianzas se ajustan con los \mathbf{W})

Anotaciones sobre gaussianas N -dimensionales

- Ejemplo: datos artificiales

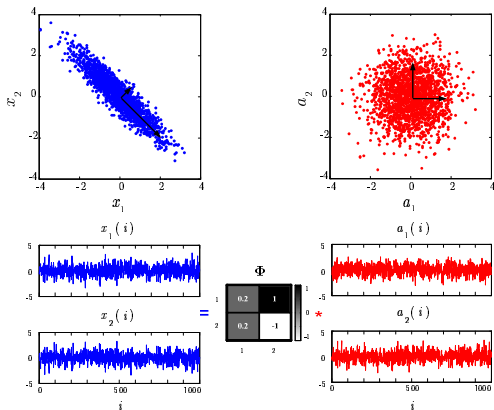


Figura: Mezcla lineal de dos fuentes no correlacionadas con distribución gaussiana ($\mathbf{U}_j = \mathbf{I}$), se convierte en el caso general (\mathbf{U}_j general).