

# Facultad de Ciencias, UNAM

## Redes Neuronales

### Tarea 2

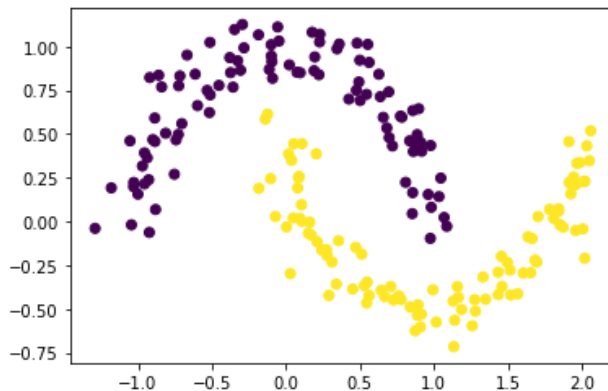
Rubí Rojas Tania Michelle

13 de abril de 202

1. Usando *sklearn.datasets.make moons* genera un conjunto de datos de la siguiente forma:

```
In [1]: C1, C2 = moons(random state=123, n samples=200, noise=0.1)
```

```
1 from sklearn.datasets import make_moons
2 import matplotlib.pyplot as plt
3
4 # Generamos un conjunto de datos.
5 C1, C2 = make_moons(random_state = 123, n_samples=200, noise=0.1)
6 plt.scatter(C1[:,0], C1[:,1], c=C2); plt.show()
7
```



- a) Implementa la regresión logística usando el descenso gradiente para clasificar  $C_1$  y  $C_2$ .
  - b) ¿Qué transformación de los datos ocupaste para poder hacer la correcta clasificación?
2. Calcula la derivada de la tangente hiperbólica  $\tanh$ .

SOLUCIÓN:

$$\begin{aligned}
 \frac{d}{dx} \tanh(x) &= \frac{d}{dx} \left( \frac{\sinh(x)}{\cosh(x)} \right) && \text{definición de } \tanh(x) \\
 &= \frac{(\sinh'(x) \cdot \cosh(x)) - (\cosh'(x) \cdot \sinh(x))}{\cosh^2(x)} && \text{derivative quotient rule} \\
 &= \frac{(\cosh(x) \cdot \cosh(x)) - (\sinh(x) \cdot \sinh(x))}{\cosh^2(x)} && \sinh'(x) = \cosh(x) \text{ y } \cosh'(x) = \sinh(x) \\
 &= \frac{\cosh^2(x) - \sinh^2(x)}{\cosh^2(x)} && \text{aritmética} \\
 &= \frac{1}{\cosh^2(x)} && \cosh^2(x) - \sinh^2(x) = 1 \\
 &= \operatorname{sech}^2(x) && \frac{1}{\cosh^2} = \operatorname{sech}^2(x)
 \end{aligned}$$

3. Usando el perceptrón multicapa visto en clase, clasifica a  $C_1$  y  $C_2$ . ¿Qué parámetros ocupaste?
4. Con la red neuronal, vista en clase, que hace la clasificación multiclase usando la función *softmax*, realiza los siguiente ejercicios:
  - a) Encuentra la mejor arquitectura para el conjunto de Iris. Justifica tu respuesta de por qué es la mejor.
  - b) Usa las funciones  $\tanh$  y  $\gamma$  en la capa intermedia. ¿Cuál funciona mejor?
  - c) Clasifica los siguientes estímulos y reporta a qué clase pertenece cada uno:
    - 5.97 4.20 1.23 0.25
    - 6.80 5.00 1.25 1.20
    - 12.50 9.20 40.32 21.55
  - d) ¿Te parecen correctas todas las clasificaciones? En caso de que alguna no, ¿por qué? ¿cómo corregirías este error?
5. ¿Qué es y cómo funciona la función de activación *Radial Basis Function (RBF)*?

SOLUCIÓN: Una *función de base radial*  $\phi(x)$  es una función que calcula la distancia euclideana de un vector de entrada  $x$  respecto de un centro  $c$ , de tal manera que resulta la siguiente función:

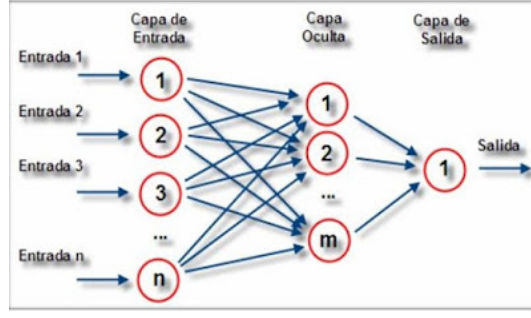
$$f(x) = (\|x - c_i\|)$$

A cada neurona de la capa de entrada le corresponde una función de base radial  $\Phi(x)$  y un peso de salida  $w_i$ . El patrón de salida ingresa a una neurona de salida que suma las entradas y da como resultado una salida. La función de una red *RBF* final resulta:

$$F(x) = \sum_{i=1}^N w_i \Phi(\|x - c_i\|)$$

Las redes *RBF* tienen una construcción rígida de tres capas:

- **Capa de Entrada.** Transmiten las señales de entrada a las neuronas ocultas sin realizar procesamiento, es decir, las conexiones de la capa de entrada a la capa oculta no llevan pesos asociados.



- **Capa Oculta.** Realizan una transformación local y no lineal de dichas señales.

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

La salida de cada elemento de la capa oculta  $z_i(n)$  se calcula como la distancia que existe entre el patrón de entrada  $X(n)$  al centro del clouster  $C_i$  ponderada inversamente por  $d_i$ , y aplicando después a ese valor una función de base radial.

$$z_i(n) = \Phi \left( \frac{\left( \sum_{j=1}^p (x_j(n) - c_{ji})^2 \right)^{\frac{1}{2}}}{d_i} \right) \quad i \in \{1, 2, \dots, m\}$$

donde  $\Phi$  es una función de base radial, dentro de éstas la más utilizada es la función Gaussiana

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

- **Capa de Salida.** Realiza una combinación de las actividades de las neuronas ocultas. Tiene la responsabilidad en la red de activación de patrones aplicados en la capa de entrada.

Cada elemento de procesamiento calcula su valor neto como una combinación lineal de las salidas de los elementos de procesamiento de la capa oculta. La función de activación y transferencia es lineal, por lo que para un patrón  $n$ ,  $X(n) = (x_1(n), x_2(n), \dots, x_p(n))$ , la salida de la red asociada a cada elemento  $k$  de la capa de salida se obtiene de la forma

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

donde los  $w_{ik}$  son los pesos asociados al elemento  $k$  de la capa oculta y el elemento  $i$  de la capa oculta, que ponderan cada uno las salidas  $z_i(n)$  del elemento de procesamiento de la capa oculta correspondiente.

El término  $\mu_k$  es un término denominado umbral y está asociado a cada uno de los elementos de procesamiento de la capa de salida.

El aprendizaje consiste en la determinación de centros, desviaciones y pesos de la capa oculta a la capa de salida. Como las capas de la red realizan diferentes tareas, se separan los parámetros de la capa oculta de la capa de salida para optimizar el proceso. De esta forma, los centros y las desviaciones siguen un proceso guiado por una optimización en el espacio de entrada, mientras que los pesos siguen una optimización sobre la base de las salidas que se desean obtener.

Los métodos de aprendizaje más utilizados son el *método híbrido* y el *método totalmente supervisado*.

## Bibliografía

- [https://www.wikiwand.com/es/RNA\\_de\\_base\\_radial#/Referencias](https://www.wikiwand.com/es/RNA_de_base_radial#/Referencias)
- [http://ele.aut.ac.ir/~abdollahi/Lec\\_3\\_NN11.pdf](http://ele.aut.ac.ir/~abdollahi/Lec_3_NN11.pdf)
- <http://dianainteligenciaartificial.blogspot.com/2015/07/redes-de-neuronas-de-base-radial.html>
- <http://www.varpa.org/~mgpenedo/cursos/scx/archivospdf/Tema5-6.pdf>