

Tarea 4: Backpropagation

Leonel Guerrero

March 4, 2023

1 Multilayer Perceptron with Backpropagation (MLP-BP)

1.1 Enunciado

Implemente su propio perceptron multicapas para que tenga una capa oculta con un numero n de neuronas en la capa oculta y k neuronas en la capa de salida. n y k son parámetros de su red (función). Usted puede emplear el lenguaje de programación de su preferencia. Entregar el código con una documentación mínima

1.2 Implementación

- Github
- Google Colab

2 Prueba sobre datos

2.1 Enunciado

Para los conjuntos de entrenamiento usado en la tarea del perceptron y en el Adaline, evalúe un perceptron multicapas. Evalúe y compare este algoritmo con los resultados obtenidos en la tarea anterior. Comente sobre su escogeria en los parámetros de aprendizaje y la arquitectura requerida para la mejor convergencia

2.2 Evaluación y comparación del algoritmo

Veamos primeramente las gráficas del error cuadrático medio para los conjuntos de entrenamiento y los conjunto de validación obtenidos de los archivos EarthSpace.csv, MedSci.csv y LifeSci.csv, Agri.csv para el perceptron multicapas con 1 capa oculta de 3 neuronas y 1 capa de salida de 2 neurona:

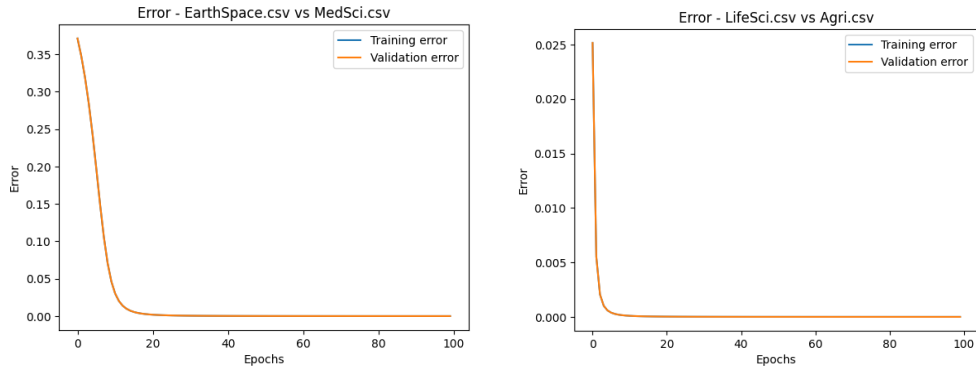


Figure 1: Error cuadrático medio para el conjunto de entrenamiento y el conjunto de validación para el perceptron multicapas con 1 capa oculta de 3 neuronas y 1 capa de salida de 2 neurona

Se puede apreciar que el error cuadrático medio para los conjuntos de entrenamiento y para los conjuntos de validación va disminuyendo a medida que se va realizando el entrenamiento. Esto es un indicador de que el algoritmo está convergiendo. De igual forma se puede ver que las curvas de error cuadrático medio para el conjunto de entrenamiento y para el conjunto de validación se encuentran muy cercanas, lo que indica que el algoritmo no está sobreajustando (overfitting) los datos.

Veamos la siguiente tabla para comparar los resultados obtenidos para el MLP contra el Adeline

	Error cuadrático medio	
	MLP	Adeline
EarthSpace vs MedSci	2.51×10^{-5}	117.6796
LifeSci vs Agri	9.21×10^{-7}	1292.4977

Se puede apreciar por medio de la tabla que el MLP obtuvo mejores resultados que el Adeline. Obteniendo un error mas pequeño para los conjuntos de entrenamiento y de validación. Sin embargo cabe destacar que los datos de entrenamiento para el MLP se le aplico una transformación en contraposición al adeline que se le pasaron los datos sin ningún tipo de tratamiento

El tratamiento realizado fue el de la normalización, en el cual se le aplico la siguiente formula a cada dato de entrada:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

Donde μ es la media de los datos y σ es la desviación estándar de los datos. Esto se realizo para que los datos de entrada tenga una distribución normal y asi poder obtener mejores resultados. La Inicialización de los pesos se realizo tomando valores aleatorias de la distribución normal con media 0 y desviación estándar 1.

Para el MLP se escogió una tasa de aprendizaje de 0.001 ya que en diferentes experimentos y tomando como referencia los pesos utilizados en el Adeline se escogió esa tasa como la que mejor resultados daba. Para la arquitectura de la red se escogió una capa oculta de 3 neuronas y una capa de salida de 2 neuronas. Esto se escogió ya que en diferentes experimentos se observo que con esa arquitectura se obtuvieron mejores resultados que con otras arquitecturas.

Aparte también se escogió un valor de momentum para aumentar la velocidad de convergencia. El valor de momentum escogido fue de 0.2. Esto se escogió ya que en diferentes experimentos se observó que con ese valor se obtuvieron mejores resultados que con otros valores de momentum.

3 Expert machine

3.1 Enunciado

Considere una máquina compuesta por K expertos. La función de entrada y salida del k -ésimo experto está dada por $F_k(x)$, donde x es el vector de entrada y $k = 1, 2, \dots, K$. Las salidas individuales de los expertos están linealmente combinadas para formar la salida general y , definida por

$$y = \sum_{k=1}^K w_k F_k(x)$$

donde w_k es el peso lineal asignado a $F_k(x)$. El requerimiento es evaluar w_k tal que y resulte ser el estimado de mínimos cuadrados de la respuesta deseada d según x . Dado un conjunto de entrenamiento $(x_i, d_i)_{i=1}^N$, determine los valores requeridos de los w_k para resolver este problema de estimación de parámetros.

3.2 Respuesta

El objetivo es encontrar los valores de los pesos w_k que minimizan el error cuadrático medio entre la salida de la máquina y la salida deseada, dada por:

$$E(w) = \frac{1}{2} \sum_{i=1}^N (y_i - d_i)^2$$

donde y_i es la salida de la máquina para la entrada x_i , y d_i es la salida deseada correspondiente.

Podemos encontrar los valores de w_k que minimizan $E(w)$ tomando la derivada de $E(w)$ con respecto a cada peso w_k y estableciéndolos en cero:

$$\frac{\partial E}{\partial w_k} = \sum_{i=1}^N (y_i - d_i) \frac{\partial}{\partial w_k} \left(\sum_{j=1}^K w_j F_j(x_i) \right) = \sum_{i=1}^N (y_i - d_i) F_k(x_i) = \sum_{i=1}^N (w_1 F_1(x_i) + w_2 F_2(x_i) + \dots + w_k F_k(x_i) + \dots - d_i)$$

Igualando esto a cero, obtenemos:

$$\sum_{i=1}^N (y_i - d_i) F_k(x_i) = 0$$

Lo que nos da la ecuación para el peso w_k :

$$w_k = \frac{\sum_{i=1}^N d_i F_k(x_i)}{\sum_{i=1}^N F_k(x_i)^2}$$

Por lo tanto, para encontrar los valores de w_k que minimizan el error cuadrático medio, debemos calcular las sumas $\sum_{i=1}^N d_i F_k(x_i)$ y $\sum_{i=1}^N F_k(x_i)^2$ para cada k , y utilizarlos en la fórmula anterior