

# Perceptrón Multicapa

García Prado, Sergio  
sergio@garciparedes.me

15 de mayo de 2017

## Resumen

*En este documento se realiza una breve descripción acerca de la Red Neuronal Artificial Perceptrón Multicapa. Además, se han realizado experimentos sobre dos conjuntos de datos, el primero de ellos con valor de destino de carácter numérico (regresión) y el segundo de carácter categórico (clasificación). En estos experimentos se ha comparado el Perceptrón Multicapa con otras estrategias como la Regresión Lineal, Regresión Logística y el ADALINE.*

## 1. INTRODUCCIÓN

### 1.1. PERCEPTRÓN MULTICAPA

El perceptrón multicapa es una red neuronal artificial formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón simple. El perceptrón multicapa puede ser totalmente o localmente conectado. En el primer caso cada salida de una neurona de la capa “ $i$ ” forma parte de la entrada de todas las neuronas de la capa “ $i+1$ ”. [Wik17]

El perceptrón multicapa está formado por tres capas principales (entrada, oculta y salida). Mientras que las capas de entrada y salida únicamente están formadas por una única capa, constituyéndose la de entrada por una neurona por atributo del conjunto de datos y la de salida por una neurona por clase de destino. Por contra, la capa oculta puede estar formada por más de una capa.

Las funciones de activación de las neuronas de las capas de entrada y salida son de carácter lineal mientras que las de las neuronas de las capas ocultas son funciones no lineales como la tangente hiperbólica o la función logística cuyos dominios de aplicación están acotados en los intervalos  $(-1, 1)$  y  $(0, 1)$  respectivamente. Dichas funciones deben ser derivables puesto que el proceso de aprendizaje se basa en la retropropagación o regla delta generalizada (una especialización del método del gradiente descendiente, que ajusta los pesos de cada neurona de forma iterativa).

## 2. EXPERIMENTOS SOBRE COMPUTER HARDWARE DATASET

En esta sección se describe el experimento realizado sobre el conjunto de datos **Computer Hardware** [UCIa] con distintos algoritmos de aprendizaje que se describen a continuación. El conjunto de datos está formado por 7 atributos normalizados junto con el valor de la clase. El conjunto utilizado para entrenamiento está formado por 139 instancias y 70 instancias para test. Puesto que el algoritmo de aprendizaje realiza una regresión sobre los datos, para calcular la tasa de fallos se ha utilizado como cota máxima de medida un **error relativo del 10%** sobre el valor esperado. Los parámetros utilizados para el los algoritmos de aprendizaje iterativos han sido un ratio de aprendizaje  $\alpha = 0,3$  y 5000 épocas.

Los algoritmos utilizados para el aprendizaje han sido *Regresión Lineal* y *ADALINE* utilizando las implementaciones de los trabajos [GPFA17b] y [GPFA17a] respectivamente. Estos resultados se han comparado con los obtenidos a partir del algoritmo de aprendizaje basado en Redes Neuronales *Perceptrón Multicapa* implementado en la biblioteca *WEKA*[too]. Se han probado distintos valores para el cardinal de la capa oculta de neuronas. Los valores utilizados han sido 10, 15, 20 y 30 neuronas.

La metodología experimental que se ha seguido ha sido un **HoldOut** con particionamiento de los datos de manera que  $\frac{2}{3}$  son utilizados en la fase de entrenamiento y  $\frac{1}{3}$  en la de test. Los resultados obtenidos tras dicha experimentación se recogen en la tabla 1.

	Computer Hardware Dataset					
	<i>R. Lineal</i>	<i>ADALINE</i>	<i>MLP 10</i>	<i>MLP 15</i>	<i>MLP 20</i>	<i>MLP 30</i>
Tasa Error HoldOut	67,143 %	78,571 %	10,0 %	14,285 %	11,428 %	5,714 %

**Tabla 1:** Tasas de error obtenidas en los experimentos sobre el conjunto de datos *Computer Hardware*

Las tasas de error obtenidas tras los experimentos muestran una tendencia a destacar. Los resultados obtenidos en los experimentos en los que el algoritmo de aprendizaje usado es el *Perceptrón Multicapa* presentan tasas de error muy inferiores a las de la *Regresión Lineal* y el *ADALINE*. Una de las razones por las cuales dichas estrategias obtienen una tasa de error tan alta puede ser debida a que los resultados no sean linealmente separables, es decir, no pueden ser divididos mediante hiperplanos. Puesto que el *Perceptrón Multicapa* utiliza funciones de activación no lineales en la capa oculta, lo cual añade la características de que la red se adapte a resultados no lineales. En cuanto al número de neuronas en la capa oculta, los mejores resultados se escogen con 30 neuronas.

### 3. EXPERIMENTOS SOBRE WINE DATASET

En esta sección se realiza un experimento para estudiar la tasa de error obtenida mediante el algoritmo de aprendizaje automático basado en redes neuronales *Perceptrón Multicapa* para después compararla con los resultados obtenidos mediante la estrategia de *Regresión Logística*. Dicha labor es posible cuando la clase de destino es numérica pero se puede discretizar. (En este caso es de carácter entero y acotada por lo que su discretización es trivial). Puesto que no es de tipo binario es necesario seguir la estrategia de clasificación por pares tal y como se ha descrito anteriormente.

Los algoritmos utilizados para el aprendizaje han sido *Regresión Logística* utilizando las implementación de los trabajos [GPFA17b]. Estos resultados se han comparado con los obtenidos a partir del algoritmo de aprendizaje basado en Redes Neuronales *Perceptrón Multicapa* implementado en la biblioteca *WEKA*[too]. Se han probado distintos valores para el cardinal de la capa oculta de neuronas. Los valores utilizados han sido 10, 15, 20 y 30 neuronas.

El conjunto de datos que se ha utilizado para dichos experimentos es **Wine Data Set** [UCIb], formado por **178 instancias**. Dichas instancias contienen **12 atributos** de carácter numérico. La **clase de destino es de carácter numérico**. Sin embargo, en este caso es de tipo entero, por lo que puede ser vista como una variable categórica. En cuanto a la metodología experimental, se ha seguido una extrategia de **HoldOut** con particionamiento de los datos de manera que  $\frac{2}{3}$  son utilizados en la fase de entrenamiento y  $\frac{1}{3}$  en la de test. Los resultados de dichos experimentos se muestran en la tabla 2.

	Wine Dataset				
	<i>R. Logística</i>	<i>MLP 10</i>	<i>MLP 15</i>	<i>MLP 20</i>	<i>MLP 30</i>
Tasa Error HoldOut	3,389 %	5,084 %	5,084 %	5,084 %	5,084 %

**Tabla 2:** Tasas de error obtenidas en los experimentos sobre el conjunto de datos *Wine*

Las tasas de error en este caso presentan un suceso curioso puesto que para todos los distintos tamaños de la capa oculta del *Perceptrón Multicapa* el resultado ha sido el mismo (por lo cual en una implementación real sería conveniente utilizar la de 10 neuronas). En este caso los resultados obtenidos a partir de la *Regresión Logística*. Una de las causas de este suceso puede ser que las clases de destino si que presenten la característica de ser linealmente separables.

## REFERENCIAS

- [CCAG17] Teodoro Calonge Cano and Carlos Javier Alonso González. Técnicas de Aprendizaje Automático, 2016/17.
- [GP17] Sergio García Prado. Perceptrón multicapa. <https://github.com/garciparedes/machine-learning-multilayer-perceptron>, 2017.
- [GPFA17a] Sergio García Prado and Óscar Fernández Angulo. Redes neuronales monocapa. <https://github.com/garciparedes/machine-learning-single-layer-neural-networks>, 2017.
- [GPFA17b] Sergio García Prado and Óscar Fernández Angulo. Regresión lineal múltiple y logística. <https://github.com/garciparedes/machine-learning-regression>, 2017.
- [too] Weka. <http://www.cs.waikato.ac.nz/ml/weka/>.
- [UCIa] UCI Machine Learning Repository. Computer Hardware Data Set. <http://archive.ics.uci.edu/ml/datasets/Image+Segmentation>.
- [UCIb] UCI Machine Learning Repository. Wine Data Set. <http://archive.ics.uci.edu/ml/datasets/Wine>.
- [Wik17] Wikipedia. Multilayer perceptron — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Multilayer%20perceptron&oldid=776866963>, 2017. [Online; accessed 15-May-2017].