

Redes Neuronales Monocapa

García Prado, Sergio
sergio@garciparedes.me

Fernández Angulo, Óscar
oscar.fernandez.angulo@alumnos.uva.es

11 de mayo de 2017

Resumen

En este documento se realiza una descripción acerca de dos de las estrategias de aprendizaje basadas en redes neuronales de una única capa (monocapa). Se describe el funcionamiento del Perceptrón Simple así como el ADALINE. Además, se realiza una implementación en el lenguaje Octave de las dos estrategias de aprendizaje para después utilizarla en dos ejemplos prácticos. El primero de ellos sobre un conjunto de datos Simple y el segundo a partir del conjunto de datos Computer Hardware[UCI].

1. INTRODUCCIÓN

1.1. PERCEPTRÓN SIMPLE

El perceptrón simple consiste en uno de los modelos de redes neuronales más simples que existen. Esta estrategia sigue el modelo descrito por *McCulloch-Pitts*. Tiene el comportamiento de un clasificador binario, puesto que puede ser visto como una función $f(x)$ siendo x el vector de dimensión n que representa las instancias a clasificar. El dominio de aplicación de la función f es el subconjunto $\{-1, 1\}$, por lo que funciona como un clasificador binario. Nótese que para extender su funcionamiento a un mayor número de clases habría que utilizar estrategias complementarias como un perceptrón simple por clase o estrategias pareadas.

```
function s = signo(x)
    s = zeros(size(x));
    for i = 1:size(x)
        if x(i) >= 0
            s(i) = 1;
        else
            s(i) = -1;
        end
    end
end
```

Figura 1: Octave: `/src/signo.m`

El motivo por el cual el dominio de aplicación es $\{-1, 1\}$ viene dado por la función $F(x)$ utilizada para la clasificación, la cual se describe en la figura 1 mediante su implementación en Octave. La tarea de aprendizaje consiste en el “acercamiento” de la función $f(x)$ al valor esperado de los datos mediante la técnica del gradiente descendente, que ajusta los pesos de manera iterativa hasta que todas las instancias del conjunto de datos de entrenamiento son clasificadas de manera correcta. Dicha estrategia se muestra en la figura 2, que representa la fase de aprendizaje.

1.2. ADALINE

La estrategia del *ADALINE* es similar a la del Perceptrón Simple de manera conceptual, sin embargo en este caso en lugar de representar un clasificador, se representa una regresión. Por lo

```
function w = simple_perceptron_k(x, y, bias, w=rand(size(x,2),1))
    f = 0;
    while f == 0
        f = 1;
        w
        for i = 1:size(x,1)
            d = signo((x(i,:) * w + bias));
            if d ~= y(i)
                w = w - (d - y(i))*x(i,:)' ;
                f = 0;
            end
        end
    end
end
```

Figura 2: Octave: `/src/simple_perceptron_k.m`

tanto, desde el punto de vista matemático, si se ve el *ADALINE* como una función $f(x)$, el dominio de aplicación en este caso está formado por R . Esto se consigue sustituyendo la función signo descrita anteriormente por la función $F(x) = x$. El aprendizaje mediante esta estrategia se basa en la *Regla Delta*, una técnica basada en el método del gradiente descendente. Por lo cuál, utiliza un aprendizaje iterativo determinado por el número de épocas y un coeficiente de aprendizaje. Esta estrategia se puede visualizar en la figura 3 mediante su implementación en *Octave*.

```
function w = adaline_k(x, y, seasons=5000, alpha = 0.1)
    w = zeros(size(x)(2), 1);
    for t = 1:seasons
        for i=1:size(x,1)
            p = (x(i,:) * w);
            w += alpha * (y(i)-p) * x(i,:)' ;
        end
    end
end
```

Figura 3: Octave: `/src/adaline_k.m`

2. PERCEPTRÓN SIMPLE SOBRE CONJUNTO DE DATOS SENCILLO

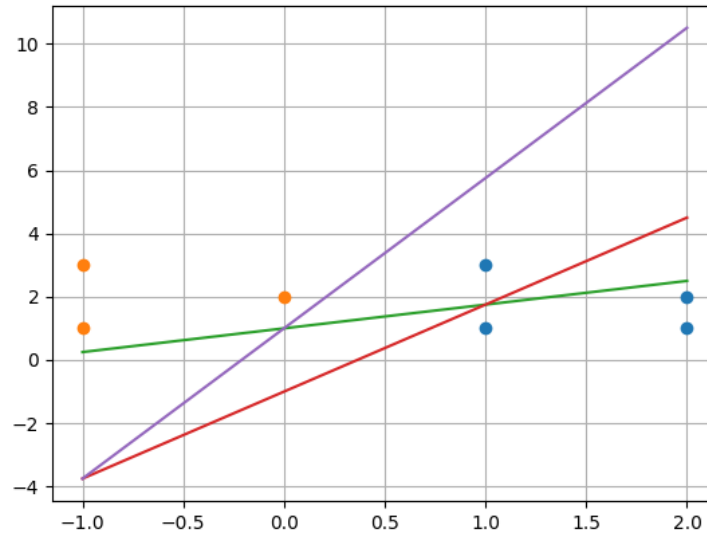
En esta sección se describe el experimento realizado utilizando la estrategia del *perceptrón simple*. El conjunto de datos utilizado se muestra en la tabla 1. Este conjunto de datos posee la característica de ser linealmente separable, tal y como se verá a continuación. Está formado por **8 instancias** formadas por **2 atributos** de carácter numérico. La clase de destino también es de carácter numérico, formada por el conjunto de valores $D = \{1, -1\}$. Puesto que este conjunto está formado por dos categorías, entonces se puede utilizar un único clasificador binario (perceptrón simple en este caso) para clasificar las instancias.

Simple Dataset		
x_1	x_2	$d(x)$
1	1	1
2	1	1
2	2	1
1	3	1
-1	1	-1
0	2	-1
-1	3	-1

Tabla 1: *Conjunto de datos Simple*

En este caso, debido al reducido tamaño del conjunto de datos, se ha realizado un experimento basado en el error de resustitución. Por lo tanto, se ha utilizado el 100 % de las instancias, tando para entrenamiento como para test. Para poder realizar comparaciones, en lugar de escoger de manera aleatoria los pesos iniciales se ha utilizado $W = [0,75, 1]$ y $bias = -3$.

Puesto que instancias del conjunto de datos son linealmente separables tal y como se ha indicado anteriormente (se puede apreciar en la figura 4) y la metodología experimental seguida es se apoya en el error de resustitución. La tasa de error obtenida es del 0,0 % tal y como se muestra en la tabla 2.

**Figura 4:** *Evolución de los pesos conforme aumenta el número de iteraciones*

La figura 4 muestra la evolución de la recta que separa las dos clases de datos conforme aumentan las iteraciones. La línea en color **verde** se corresponde con la **iteración inicial**, la línea en color

verde se corresponde con la **primera iteración** y, por último, la línea **púrpura** se corresponde con la **segunda iteración**. Nótese que en este punto las dos clases están separadas por dicha recta, por lo que el algoritmo ya está en condiciones de terminar.

Perceptrón Simple - Simple Dataset	
Error de Resubstitucion	0,0 %

Tabla 2: Resultados del experimento sobre el conjunto de datos Simple

3. ADALINE SOBRE COMPUTER HARDWARE DATASET

En esta sección se describe el experimento realizado sobre el conjunto de datos **Computer Hardware**[UCI] y el algoritmo **Adaline**. Puesto que el algoritmo de aprendizaje realiza una regresión sobre los datos, para calcular la tasa de aciertos se ha utilizado como unidad de medida un **error relativo del 10 %** sobre el valor real. Los parámetros utilizados para el ADALINE han sido un ratio de aprendizaje $\alpha = 0,1$ y 5000 épocas.

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 3. Tal y como se puede apreciar, la tasa de error obtenida es muy elevada. Esto puede ser debido a la existencia de instancias muy diferentes unas de otras, lo cual penaliza drásticamente los resultados.

Adaline — Computer Hardware Dataset	
Error HoldOut	71,429 %

Tabla 3: [TODO]

Tras realizar un experimento que utilizara todo el conjunto de datos (tanto para entrenamiento como para test) el error obtenido se redujo a 0,0 %. Esta tarea se ha llevado a cabo para comprobar la corrección de la implementación utilizada. Sin embargo, nótese que esta circunstancia no es significativa para comprobar la validez del clasificador ADALINE sobre el conjunto de datos Computer Hardware puesto que en un entorno real esta situación no es posible.

REFERENCIAS

- [CCAG17] Teodoro Calonge Cano and Carlos Javier Alonso González. Técnicas de Aprendizaje Automático, 2016/17.
- [UCI] UCI Machine Learning Repository. Computer Hardware Data Set. <http://archive.ics.uci.edu/ml/datasets/Image+Segmentation>.