

## Proyecto 5

Ruben Cuadra

Dr. Victor de la Cuva

Aprendizaje Automatico

October 11, 2017

### Backpropagation en una Neurona

Hacer una función que logre hacer clasificación de dos clases. La función implementa una red neuronal feedforward de una sola neurona, con entrenamiento Backpropagation, mejor conocida como BPN.

El código consta de una librería llamada Proyecto5.py la cual consta de 10 funciones, todas giran entorno a la función: (Las negritas son los parámetros que recibe)

*bpnUnaNeurona:*

**w** : Vector de pesos

**layerSize** : Número de capas, no se usa

**Y** : Matrix con valores en X

**Y** : Vector con valores en Y, para cada renglón en X

**alpha** (opcional) : Razón de aprendizaje , por default es 0.01

**e**: Minimo error valido, default es 0.01

**activacion**: Es un enumerator que proviene de *activaciones* , los validos son *activaciones.LINEAL* y *activaciones.SIGMOIDAL*

**iteraciones** (opcional): Default es 100

Nos devuelve 2 vectores, el primero es un vector de pesos donde el elemento 0 representa el valor *b* y el segundo vector es del tamaño *iteraciones*

Para su correcto funcionamiento usa las funciones:

- *getCostFunction:*

Recibe la variable *activacion* (misma que recibe *bpnUnaNeurona*) y nos regresa una función, la función que devuelve recibe 4 parámetros y devuelve un valor numérico, recibe :

**weights** : Vector con valores de pesos de tamaño 1xM

**w0** : Valor numérico que representa el peso *b*

## Proyecto 5

**X** : Matrix con entradas X donde cada renglón es un ejemplo, tamaño NxM

**Y** : Vector con resultados para cada ejemplo X, su tamaño es 1xM

La función regresa el valor numérico que representa el costo de esos pesos/ejemplos/Y

### - *getAFunction*:

Recibe la variable *activacion* (misma que recibe *bpnUnaNeurona*) y nos regresa una función, la función que devuelve recibe 1 parámetro :

**z** : Vector con resultados de la evaluación de la hipótesis, si existen N ejemplos z tendrá tamaño 1xN

La función devuelta recibe esta z y devuelve un vector con mismas dimensiones

La función regresa el valor numérico que representa el costo de esos pesos/ejemplos/Y

### - *getDZFunction*:

Recibe la variable *activacion* (misma que recibe *bpnUnaNeurona*) y nos regresa una función, la función que devuelve recibe 2 parámetro :

**A** : Vector con resultados de la evaluación de la función A sobre un vector Z

**Y** : Vector con valores para cada ejemplo en X, mismo tamaño que A

La función devuelta regresa un vector del mismo tamaño que A que es del mismo tamaño de Y

### *prediceRNYaEntrenada*:

**X** : Matrix de valores en X, Tamaño NxM

**weights** : Vector W, su tamaño es 1x(N+1), donde el elemento 0 representa la *b* de pesos

**activationFunction** : es una función que recibe como parámetro un vector X que representa un ejemplo de la matriz, el primer elemento de este vector siempre sera 1, en este punto el vector tiene dimensiones 1xN ; El segundo parámetro es un vector de pesos con dimensiones 1xN, al final la función debe devolver un valor numérico.

## Proyecto 5

### Requisitos

Libreria matplotlib (Graficas)

Libreria numpy (Operaciones matemáticas)

Python 2.7 o 3.5 (Instalar las librerías correctamente usando pip o pip3)

En el repositorio se encuentra un archivo ejemplo que es parseado por la función *getDataFromFile* la cual recibe como parámetro un archivo(su ruta absoluta) y nos devuelve la matriz X y el vector Y, los cuales ya pueden ser usados para todas las funciones previamente descritas **datos.csv** contiene valores numéricos x,y ; Son N columnas separadas por comas donde la ultima representara las Y y las demás valores en X.