

Proyecto 6

Perceptrón y Adaline

Descripción

Hacer dos pequeños programas en Python que implementen un Perceptrón y una Adaline para resolver el problema del AND y del OR (aunque se deben hacer generales para resolver cualquier problema). Cada programa recibe como entrada un conjunto de ejemplos (4) de la forma (entradas, salida), donde la salida es el valor de verdad del ejemplo correspondiente, y regrese los valores de los pesos ($w_0 \dots w_n$) de cada ANN necesarios para responder correctamente a una entrada dada. Estos vectores se deben calcular usando el algoritmo de aprendizaje definido para cada caso. El vector de pesos es equivalente al vector Theta en las tareas anteriores.

Datos

Los datos las tablas de verdad de las funciones lógicas AND y OR, en donde usa datos binarios. Entonces, se deberán generar 2 matrices:

1. El vector **X** de 4 X 2, conteniendo TODAS las entradas. Se refiere a los valores de las dos variables de entrada.
2. El vector **y** de 4 X 1, conteniendo TODAS las salidas. Se refieren al valor de verdad de la operación correspondiente.

Funciones

Se deberán programar 3 funciones para el Perceptrón y 3 para el Adaline.

1. **funcionCostoPerceptron(theta,X,y)**. Recibe el vector de entrada X, el de salida y, y un vector theta. Debe regresar la función de costo **J** y el gradiente **grad**. Es decir, debe regresar las variables **[J,grad]**.
2. **entrenaPerceptron(X, y, theta)**. Implementa el entrenamiento del perceptrón. La función debe regresar pesos del perceptrón en una matriz $\Theta \in \mathbb{R}^{N+1}$, donde N es el número de pesos de la neurona. El vector **y** tiene etiquetas {0,1} que corresponde al valor del verdad de la función lógica para la que se está entrenando la ANN.
3. **predicePerceptron(theta, X)**. Recibe un vector **theta** y un vector **X** para varios entradas lógicas. Regresa el vector **p** de predicción sobre el valor de verdad de las operaciones correspondientes. Si se prueba con el mismo vector **X** de entrenamiento debe tener una exactitud del 100% (porque es un ejemplo muy pero muy simple).
4. **funcionCostoAdaline(theta,X,y)**
5. **entrenaAdaline(X, y, theta)**
6. **prediceAdaline(theta, X)**

Análisis

Vaya guardando el valor del error total para cada paso del perceptrón y de la adaline. Grafíquelo y comente al respecto en la documentación.

Entrega y Revisión

- Salve su archivo con el nombre **Proyecto6.py**. Tiene que ser con ese nombre porque es el que se va a usar para la revisión automática.
- Suba sus archivos dos archivos, el del proyecto (.py) y el de la documentación (.docx o .pdf) **por separado**, NO los ponga en ZIP.
- Para la revisión se harán pruebas sobre las funciones solicitadas. Para esto, tanto el archivo que contiene las funciones (el suyo) como el de prueba (el mío), se colocarán en el mismo folder. El programa de prueba al inicio hará un:

```
from file import function
```

y luego se utilizará en el script de prueba llamándola como como:

```
function(<parámetros>)
```