

## ALGORITMO BACKPROPAGATION

Inicialmente, tenemos que generar una red neuronal para poder realizar el algoritmo BackPropagation el cual consta de ciertas funciones para poder realizar su función común y corriente.

El código adjunto a este documento que se encuentra dentro de una carpeta llamada “Perceptron” el cual es funcional para el siguiente ejemplo.

Tenemos un código .txt el cual entrena la red neuronal para el caso ideal donde recibe dos numero  $[x1, x2]$  y su resultado sera  $[1]$  cuando  $x1$  sea mayor que  $x2$ , en caso contrario su resultado sera  $[0]$ .

Explicando en que consiste el algoritmo BackPropagation, es un método de cálculo de gradiente usado en algoritmos de aprendizaje supervisado para entrenar redes neuronales artificiales. El método emplea un ciclo propagación y adaptación de dos fases, Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

Adicional a esto, tenemos que entender el funcionamiento del perceptron, el cual se encuentra en el siguiente documento “Perceptron” el cual se encuentra en la misma carpeta.

Explicando mas a fondo el código que hemos realizado, consta de una clase principal la cual se llama Perceptron ()

```
1  import math, random, os
2
3
4  > class Perceptron(): ...
132
133
134
135  if __name__ == "__main__":
136      Perceptron()
137
```

La cual consta de distintos atributos tales como:

- `Build()`: esta función de la clase se encarga de construir la red neuronal, recibiendo de entrada la cantidad de neuronas que tienen la capa de entrada y la capa de salida, dándoles pesos aleatorios.
- `Percep()`: esta función se encarga de calcular la salida de la red neuronal a partir de las entradas, en pocas palabras es el funcionamiento del perceptron el cual anteriormente ya mencionado se encuentra en otro documento adjunto en la carpeta.
- `Train()`: unicamente recibe la cantidad de veces que queremos entrenar la red neuronal. La red neuronal se entrena a partir de un documento plano el cual consta de X lineas, en este caso tenemos dos documentos planos los cuales constan de 200 lineas cada uno, el entrenamiento puede ser de 0 a 200 veces.
- `Readlines()`: se encarga de leer cada linea del documento de texto plano para entrenar la red neuronal.
- `Backpropagation()`: se encarga de realizar la corrección de errores y la propagación de los nuevos pesos
- `Save()`: guarda todo el entrenamiento realizado en un documento llamado `perceptron.txt` mediante el cual la red estará entrenada para cualquier caso
- `__init__()`: se encuentra el orden con el que se llaman todas las funciones, adicional a esto se encuentra un factor `alpha` el cual establece que tanto aprende la red neuronal o no, adicional a esto la dirección donde se guardaran y importaran los archivos de texto plano.