Proyecto Final – Parte 1
Mineria de Datos
Dr. Javier García García

Entregable 1-B

Estimación de emisiones de CO2

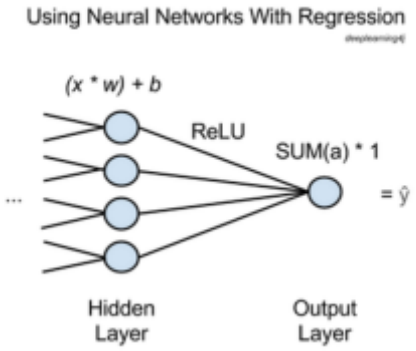
Redes neuronales

Por otro lado, ademas del algoritmo EM, decidimos tratar el problema usando el conjunto de datos como un comjutno de entrenamiento para una red neuronal que nos permita usar la regresión lineal para predecir el comportamiento de las particulas de CO_2 en distintas regiones.

El funcionamiento de las redes neuronales es bantante sencillo, se tiene un conjunto de pesos representados en forma vectorial que representan la suceptibilidad de la red ante ciertas entradas, de esta forma, se inicializan con valores aleatorios los pesos y se busca encontrar la funcion que minimize el error de estos pesos con respecto a los que debería llevar para igualar esperados a priori (los valores del conjunto de entrenamiento), de esta forma, podremos usar optmización por descenso de gradiente para ir reduciendo el error hasta encontrar un punto de convergencia.[1]

Un punto importante de las redes neuronales, son las capas que se usarán para resolver el problema, dado que se busca imitar el comportamiento de un cerebro biologico, una neurona sola (llamada perceptrón) no es suficiente para encontrar los valores que estamos buscando (los parametros del modelo que nos permite predecir el comportamiento dado un conjunto de entrenamiento), por esta razón es necesario aumentar la complejidad de la red utilizando capas de redes para encontrar los valores adecuados para resolver el problema.

Con cada iteración del metodo de optimización se calcula el error entre los pesos actuales de la red, buscando cambiar aquellos que generan el mayor error, de esta forma, en la salida de las multiples capas de la red neuronal, tendremos los valores de los parametros para aproximar nuestro modelo en una regin distinta.



Preparación de los datos

Afortunadamente nuestro conjunto de datos esta bastante limpio, dada eso, nos conviene solo eliminar el ID de las calles de OpenMap, así, terminaremos entrenando nuestra red neuronal con 3 atributos, latitud, longitud y algura, por esta razón, utilizaremos 3 capas ocultas en nuestro modelo con 400 perceptrones cada uno.

Detalles de implementación.

Utilizaremos *deeplearning4j* ya que nos brinda una biblioteca completa de herramientas para el modelado de redes neuronales con multiples tecnicas de optmización, además de que permite ser usado a la par de spark, pues corre sobre el JVM lo que permite ser usado en Scala.

[1] https://deeplearning4j.org/linear-regression.html