GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO DE FIN DE GRADO

Predicción de la contaminación atmosférica mediante redes neuronales artificiales

---

Anexo IV

Documentación técnica de programación

Ismael Mira Hernández

4 de julio de 2022

Índice

Introducción

En este documento se va a presentar el código fuente del sistema, tanto de la parte relacionada con el modelo de predicción creado a partir de la red neuronal como de la parte de la interfaz web.

En una metodología ágil sí que existe documentación, la documentación debe ser la útil y necesaria para permitir el mantenimiento del sistema.

Para ambos, se especifican los métodos y se describen brevemente, siguiendo la premisa de la metodología ágil de que la documentación debe ser la necesaria para permitir el mantenimiento del sistema.

Adicionalmente, se incluyen enlaces a las bibliotecas de código usadas a lo largo del trabajo para acceder a la documentación de las mismas si fuese necesario.

Código Fuente

Modelo de predicción

Partiendo de 22 valores de entrada, que se pueden recibir como atributos o bien escribirlos manualmente, se llama al método *neuralNetwork*(). Este el encargado de llamar al resto de capas y devolver un vector de 35 float que son las salidas de la red neuronal.

Resumen de métodos	
neuralNetwork	A partir de las entrada, llama a las funciones de las capas sucesivamente (*feed-forward*) hasta obtener las salidas.
scalingLayer	Escala las variables recibidas como entrada y devuelve el vector con las salidas.
perceptronLayer1	Utilizando la función de la tangente hiperbólica, simula la primera capa de perceptrón con 10 neuronas.
perceptronLayer2	Utilizando la función lineal, simula la segunda capa de perceptrón utilizada para tener 35 salidas.
unscalingLayer	Llama a la API de contaminación y recibe un JSON con los datos actuales.

Tabla 1. Métodos del modelo.

Despliegue continuo

Solamente ejecuta un *main,* sin métodos. Partiendo de un XML con la red neuronal, recibe como argumento las 22 variables de entrada y escribe a un CSV las 35 variables de salida.

Entrenamiento continuo

Solamente ejecuta un *main*, sin métodos. Partiendo de un CSV con el histórico de datos y un XML con la red neuronal, crea otro fichero XML con la red neuronal reentrenada, para poder utilizar esta nueva en el despliegue continuo.

Interfaz web

Métodos divididos en dos archivos JavaScript: el script del servidor carga la vista, obteniendo previamente los datos de las API y calculando las salidas del modelo. El script del cliente se encarga de la funcionalidad de los botones.

El archivo principal donde se encuentra la mayoría de la funcionalidad del sistema es *index.js*, encargado de cargar la vista *index.pug*.

Resumen de métodos	
getYesterdayWeatherData	Llama a la API meteorológica y recibe un JSON con los datos del día anterior.
getTodayWeatherData	Llama a la API meteorológica y recibe un JSON con los datos del día actual.
getPollutionData	Llama a la API de contaminación y recibe un JSON con los datos actuales.
getDates	Recibe dos fechas (inicio y fin) y devuelve un array con todas las fechas entre una y otra.
executeModel	Recibe las variables de entrada para el modelo y ejecuta el .exe creado a partir del modelo.
calculateOutputs	Lee el archivo CSV creado por executeModel(inputs) y obtiene las variables de salida para poder usarlas en las gráficas.
appendTodayData	Compara la fecha actual con la última del histórico de datos, y si no son iguales añade los datos de hoy al histórico, además de llamar al entrenamiento continuo.
trainNeuralNetwork	Ejecuta el .exe dedicado a reentrenar la red neuronal, ya que ha cambiado el conjunto de datos por lo que se busca actualizar los pesos y sesgos.
toggleX	Cinco métodos, siendo X el contaminante elegido en cada caso. Muestra la gráfica del correspondiente, cambiando el estilo del botón para señalar el marcado.

Tabla 2. Métodos de la interfaz web.

Documentación de bibliotecas externas

OpenNN: https://www.opennn.net/documentation/reference/annotated.html

Axios: https://axios-http.com/docs/api_intro

Charts.js: https://www.chartjs.org/docs/2.7.3/