

RESUMEN

La contaminación atmosférica es uno de los grandes problemas a los que el mundo contemporáneo tiene que hacer frente. Por ello, existe una creciente necesidad desde, principal pero no exclusivamente, las grandes ciudades de conocer la evolución de los niveles de calidad del aire y poder anticiparse en la toma de decisiones.

Para esta tarea, la inteligencia artificial es de gran utilidad en la creación de una estimación que intente mejorar la obtenida por medios tradicionales como modelos teóricos. Concretamente, técnicas de *Deep Learning* y sus aplicaciones como las redes neuronales artificiales, debido entre otras cosas a su capacidad de procesamiento en paralelo, son idóneas para este cometido.

En este trabajo vamos a obtener un conjunto de datos sobre la ciudad de Madrid para crear un modelo con el que predecir los valores de diferentes contaminantes a lo largo del tiempo integrándolo en una aplicación web para la visualización de los resultados.

Palabras clave: redes neuronales, contaminación atmosférica, Deep Learning.

ABSTRACT

Air pollution is one of the major problems that the contemporary world has to face. Therefore, there is a growing need from, mainly but not exclusively, large cities to know the evolution of air quality levels and to be able to anticipate in decision making.

For this task, artificial intelligence is very useful in the creation of an estimate that tries to improve those obtained by traditional means such as theoretical models. Specifically, Deep Learning techniques and their applications such as artificial neural networks, due, among other things, to their parallel processing capacity, are ideal for this task.

In this work we will obtain a set of data about the city of Madrid to create a model to predict the values of different pollutants over time by integrating it into a web application for the visualization of the results.

Keywords: neural networks, air pollution, Deep Learning.

INTRODUCCIÓN

Contexto

Desde la Revolución Industrial, los niveles de contaminación han ido aumentando enormemente, pero no ha sido hasta hace pocos años cuando la sociedad ha comenzado a tomar verdadera conciencia del asunto y los gobiernos han empezado a implantar medidas, quizás viéndose obligados por los malos pronósticos que la comunidad científica auguraba sobre el futuro de nuestro planeta a bastante corto plazo.

Este aumento de la polución se ha reflejado tanto en la aparición de nuevos problemas como en el agravante de algunos existentes, en cuestiones tan diversas como el aumento de las complicaciones respiratorias, la desaparición de especies animales y vegetales, el aumento de la temperatura o la acidificación de los océanos.

La lucha contra el cambio climático, en cualquiera de sus distintas facetas, es uno de los grandes desafíos a los que el mundo va a tener que afrontar las próximas décadas. Concretamente, la contaminación ambiental, definida como *“la presencia de componentes nocivos (ya sean químicos, físicos o biológicos) en el medio ambiente (entorno natural y artificial), que supongan un perjuicio para los seres vivos que lo habitan, incluyendo a los seres humanos”* <https://www.bbva.com/es/sostenibilidad/que-es-y-que-tipos-de-contaminacion-ambiental-existen/> , conlleva un riesgo para la salud humana y para la del planeta en general.

Además de la concienciación individual necesaria para intentar frenar este proceso, las administraciones públicas deben aplicar normas para disminuir la concentración de sustancias contaminantes en la atmósfera. Esto conlleva un análisis previo de los datos disponibles (recogidos a través de estaciones, similar al proceso seguido con los datos meteorológicos) para garantizar que las decisiones tomadas son las correctas y el grado con el que se está actuando es el adecuado, tanto a corto como a largo plazo.

Estas mediciones que realizan distintos organismos a lo largo de todo el mundo también pueden ser utilizadas en la predicción de los valores futuros de los contaminantes. Desde hace varias décadas existen modelos teóricos que, basándose en la física, intentan simular el comportamiento de un sistema para obtener tendencias, pero usando la capacidad de procesamiento de los computadores actuales además de los avances en inteligencia artificial se busca mejorar estas predicciones.

En nuestro caso, aplicando técnicas de Machine Learning y Deep Learning usaremos los datos provistos por las agencias para crear modelos que permitan predecir con el mínimo error posible los niveles de contaminación de fechas venideras, previniendo episodios de alta contaminación y las consecuencias drásticas que estos conllevan.

Contaminantes principales y límites

Existen gran cantidad de sustancias nocivas presentes en el aire, aunque la EPA (Agencia de Protección Ambiental de Estados Unidos <https://www.cdc.gov/air/pollutants.htm>) identifica seis contaminantes como los principales, regulados mediante valores límite basados en los efectos que provocan tanto para la salud pública como para el medio ambiente. Son el monóxido de carbono, el plomo, los óxidos de nitrógeno, el ozono troposférico, la materia particulada y los óxidos de azufre. Se pueden clasificar en primarios o secundarios dependiendo si son generados por procesos humanos/naturales o generados por reacciones químicas de los primarios.

Para este trabajo, vamos a recoger los datos de los que más influencia tienen en Madrid para posteriormente realizar la predicción de cada uno de ellos. El límite marcado para cada uno de ellos viene dado por la Comunidad de Madrid (http://gestiona.madrid.org/azul_internet/html/web/2_3.htm?ESTADO_MENU=2_3), pudiendo ser horarios/diarios (no pueden superarse más de un determinado número de veces al año) o anuales.

Contaminante	Tipo	Límite
PM _{2.5}	Primario y secundario	25 µg/m³ (media anual)
PM ₁₀	Primario y secundario	40 µg/m³ (media anual)
O ₃	Secundario	120 µg/m³ (máxima diaria)
NO ₂	Primario y secundario	40 µg/m³ (media anual)
SO ₂	Primario	125 µg/m³ (media diaria)

<https://www.comunidad.madrid/servicios/salud/calidad-aire-salud>

Materia particulada: uno de los indicadores más comunes. Son una mezcla de partículas, tanto sólidas como líquidas, suspendidas en el aire. Estudios realizados en la Unión Europea (<https://www.eea.europa.eu/es/themes/air/intro>) muestran que son el agente

contaminante más nocivo para las personas, debido a la capacidad de penetrar en el cuerpo. Dependiendo del grosor, se pueden clasificar en varios tipos. Los que mediremos son:

PM_{2.5}: las partículas con diámetro menor a 2,5 micras. Son las más dañinas para la salud debido a que pueden llegar al torrente sanguíneo, lo que provoca efectos en el sistema cardiovascular además del respiratorio. Están formadas por elementos más tóxicos procedentes principalmente del tráfico urbano, y son capaces de mantenerse en el aire por más tiempo que las de mayor tamaño.

PM₁₀: las partículas con diámetro menor a 10 micras. Pueden ser inhaladas por el sistema respiratorio, aunque no atraviesan los alveolos pulmonares como las anteriormente mencionadas. El tiempo de permanencia es menor, de horas en vez de días. El 77,9% (<https://prtr-es.es/particulas-pm10,15673,11,2007.html>) de emisiones proceden del polvo resuspendido existente en la atmósfera.

O₃: el ozono troposférico es aquel presente a nivel de suelo. Aunque el ozono es fundamental para proteger de la radiación UV en la estratosfera, en niveles más cercanos la tierra es muy perjudicial. Causa problemas de salud entre los que se encuentran dificultades respiratorias, daño a los pulmones y asma. También tiene efectos adversos sobre la vegetación, interfiriendo con el proceso de la fotosíntesis.

NO₂: el principal contaminante entre los óxidos de nitrógeno. Producido por fábricas, vehículos y quema de residuos. Es un gas tóxico que además incrementa los niveles de PM_{2.5}. Tiene gran influencia sobre enfermedades respiratorias y provoca la lluvia ácida, con severos efectos sobre la fauna y flora.

SO₂: el principal contaminante entre los óxidos de azufre y el más peligroso. La mayor fuente de emisión son las fábricas industriales. También incrementan los niveles de materia particulado al igual que los óxidos de nitrógeno. Sobre las personas, afectan principalmente al sistema respiratorio. Sobre el medio ambiente, provocan una disminución en el crecimiento de plantas y árboles.

AQI (Air Quality Index)

Es un índice utilizado por agencias gubernamentales para facilitar el acceso a los datos de contaminación y marcar un criterio común, así no teniendo que utilizar las

concentraciones reales de los contaminantes y sus respectivos límites. En la Unión Europea no se usa un índice, sino que cada país tiene su propio criterio para comunicar sus niveles, mientras que por ejemplo en Estados Unidos la Agencia de Protección del Medio Ambiente establece una escala de 0 a 500 con un código de colores.

(<https://www.airnow.gov/aqi/aqi-basics/>)

Valor del índice de la calidad del aire	Amenaza para la salud	Colores
0 a 50	Buena	Verde
51 a 100	Moderada	Amarillo
101 a 150	Insalubre para grupos sensibles	Naranja
151 a 200	Insalubre	Rojo
201 a 300	Muy insalubre	Morado
301 a 500	Peligrosa	Granate

En este trabajo vamos a utilizar valores del índice de la calidad del aire en lugar de valores absolutos, utilizando el AQI estadounidense, tanto en la recogida de datos pasados, actuales y en la propia predicción de los valores futuros.

Inteligencia artificial y redes neuronales

El modelo de nuestro sistema se crea utilizando redes neuronales artificiales. Son una de las más importantes técnicas dentro del mundo de la inteligencia artificial y supusieron una revolución en este campo, difundiéndose enormemente por su capacidad para dar buenos resultados en ámbitos diversos.

AÑADIR IMAGEN, EJ NEURAL NETWORKS TUTORIAL 1

A partir de unos datos de entrada, la red neuronal los procesa y se crea un modelo, de los que hay varios tipos. Los más comunes son los de aproximación, clasificación y predicción. En este caso, por razones obvias, tendremos un modelo de predicción, usados para pronosticar el estado futuro de un sistema a partir de observaciones pasadas sobre él

mismo. Otros trabajos de predicción con redes neuronales incluyen pronósticos de ventas, macroeconómicos o de acciones de marketing.

Las características de la red neuronal influyen en el buen funcionamiento del modelo, siendo el mayor éxito posible la generalización a otros problemas. En nuestro caso, un ejemplo sería obtener buenos resultados en la ciudad de Barcelona a partir del modelo construido para Madrid. Para ello existen distintas topologías de redes neuronales, que pueden adecuarse más o menos a las características que buscamos, por lo que se entrará en detalle y experimentará con la finalidad de conseguir un error lo más pequeño posible en los valores alcanzados.

OBJETIVOS

El objetivo principal del trabajo es lograr obtener una predicción fiable del AQI para cada uno de los cinco contaminantes escogidos para la ciudad de Madrid.

Por lo tanto, es parte fundamental minimizar el error del modelo para conseguir la mayor precisión posible en los valores de salida. Para ello, se construirá la red neuronal adecuada, ajustando correctamente los parámetros y utilizando los algoritmos existentes convenientemente.

Para esta misión, es imprescindible extraer los datos históricos con exactitud desde fuentes fiables, ya que se utilizarán en el entrenamiento de la red neuronal y las posteriores pruebas. Se realizará un tratamiento de estos para pasar de unos datos sin procesar a unos que se puedan utilizar en el dominio que nos atañe, ordenándolos, encontrando incongruencias y completándolos.

Además de ellos, los datos en tiempo real también son necesarios para poder adaptar el modelo lo máximo posible y que así no haya demora en el sistema. Se escogerá la fuente de datos adecuada que nos haga contar con la mayor disponibilidad y sencillez posibles.

Un objetivo secundario de este trabajo es permitir a un usuario sin conocimientos expertos en el tema poder visualizar los valores obtenidos con el modelo de manera sencilla e intuitiva. De esta manera, personas ajenas totalmente a la inteligencia artificial pueden ver la información con una capa de transparencia que oculta toda la parte más tecnológica.

Otro objetivo adicional del sistema es mejorar a los modelos físicos ya existentes. Se ha demostrado que el Machine Learning puede mejorar a técnicas tradicionales en disciplinas como el análisis de consumidores, el conteo de imágenes o la optimización de tiempos, por lo que puede que en este campo también. Se analizarán los resultados para compararlos con otras formas de predicción de la contaminación del aire y sacar conclusiones.

Por último, además de contar con el despliegue continuo prediciendo los valores de contaminación siempre para la siguiente semana y no para una semana en concreto, un objetivo secundario es integrar lo conocido en el mundo de la inteligencia artificial como “continuous training” (<https://omdena.com/blog/continuous-training-machine-learning-models/>) . Actualizando los datos históricos con los nuevos que se obtienen cada día, se busca entrenar la red neuronal cada cierto plazo para obtener modelos perfeccionados.

Con esto se busca que nuestro modelo se transforme a lo largo del tiempo, ya que si esto no ocurre puede quedarse obsoleto tras cierto tiempo, sobre todo con las grandes alteraciones que provoca el cambio climático.

CONCEPTOS TEÓRICOS

Conjunto de datos <https://www.neuraldesigner.com/learning/tutorials/data-set>

La información necesaria para el entrenamiento de la red neuronal está contenida en lo que se conoce como dataset. El formato más común es CSV, que es el que se usará en este trabajo. Dependiendo del ámbito donde se quiera utilizar, otros formatos como SQL pueden ser más adecuados. Se utiliza un modelo de tabla donde las columnas son las variables mientras que las filas son las muestras.

Las variables tienen varias clasificaciones posibles. Por un lado, en función de su uso, pueden ser de entrada, de salida o no utilizadas. Por otro lado, en función de su tipo pueden ser numéricas, binarias, categóricas y no utilizadas.

Uso

Variables de entrada: conocidas como atributos. Desde un punto de vista matemático, son las variables independientes. En nuestro contexto, estas variables de entrada serán series temporales, que representan el histórico de muestras de la ciudad de Madrid.

Variables de salida: conocidas como etiquetas. Desde un punto de vista matemático, son las variables dependientes. En un problema de predicción como el que se expone en este trabajo, son los valores que queremos obtener, utilizando las variables de entrada necesarias.

Variables no utilizadas: puede haber columnas que no se usen en la construcción del modelo ya que no aportan información adicional, y en cambio incrementan la complejidad del sistema. Un buen ejemplo son columnas que contengan siempre valores constantes.

Tipo

Variables numéricas: pueden contener cualquier número tanto positivo como negativo, además de ser decimales. En nuestro sistema son las únicas que usaremos, excepto variables de fecha al ser nuestro dataset una serie temporal.

Variables binarias: solo pueden tomar dos valores. Normalmente son traducidas a variables numéricas que toman 0 y 1.

Variables categóricas: similares a las variables binarias, pero pueden tomar más de dos valores. Habitualmente son traducidas a variables numéricas donde cada posible valor es un número.

Variables de fecha: indican cualquier información relativa a la referencia temporal de la muestra, como puede ser el día, mes, año, día de la semana... Se usarán en la predicción de tendencias y como información adicional.

Transformación del dataset AÑADIR TABLAS CON EJEMPLOS CONCRETOS PARA ESTE PROBLEMA, QUIZA MEJOR EN ASPECTOS RELEVANTES

Es importante indicar que, debido a que nuestro conjunto de datos son series temporales, es necesario hacer una serie de transformaciones para adaptarlo al entrenamiento de una red neuronal. Esto ocurre debido a que en una muestra del conjunto de datos no existen variables de salida, solamente variables de entrada, por lo que hay que definir qué variables son las que se quieren predecir.

Para este cambio de formato, se introducen los conceptos de lags y steps ahead, que serán dos números. El número de lags indica el número de elementos anteriores que se quiere tener como variable de entrada en una muestra. Para este trabajo, como una muestra indica un día en concreto, si el número de lags fuera, por ejemplo, 2, significa que en la muestra de hoy aparecen las series temporales de ayer y de antes de ayer.

Por otro lado, el número de steps ahead indica el número de elementos que se van a tener como variables de salida en una muestra. En este trabajo, si por ejemplo solo quisiéramos predecir los valores para el día siguiente el número de steps ahead sería 1.

Por lo tanto, van a cambiar las dimensiones del conjunto de datos, pasando de (n° filas x n° columnas) a $(lags * steps_ahead - 1) \times (n^\circ \text{entradas} * lags + n^\circ \text{salidas} * steps_ahead)$.

La elección de estos dos valores es importante en el desarrollo correcto de un modelo. Aunque elegir el número de steps ahead es sencillo debido a que representa el número de muestras futuras que se quiere predecir, el de lags no es tan sencillo de escoger ya que no tiene relación directa con los resultados obtenidos, por lo que es necesario realizar pruebas para comprobar qué número de lags es el que más se adecúa al modelo en cuestión a construir.

Por último, el dataset transformado se divide en tres secciones: dataset de entrenamiento, dataset de selección y dataset de validación. El de entrenamiento se usa para probar diferentes modelos con distintas características y así construir el modelo final, que es la función del de selección, ya que elige los que mejores resultados han dado, y por último el de validación prueba el modelo conseguido con muestras no utilizadas en el proceso de entrenamiento con lo que comprueba la validez de este.

Habitualmente, el de entrenamiento corresponde al 60% del dataset completo mientras que el de selección y validación al 20% cada uno. En cualquier caso, para nuestro sistema puede ser que otros porcentajes nos den mejores resultados, por ejemplo, incrementando las muestras usadas en el entrenamiento y validando sólo para el porcentaje correspondiente a 365 muestras, ya que generalmente si se obtienen buenos datos para un año en concreto ocurrirá lo mismo con el resto de los años.

A partir de este momento ya es posible trabajar correctamente con el conjunto de datos procesado, pudiendo realizar tareas que serán útiles en el diseño del modelo como el cálculo de correlaciones entre las variables, el tratamiento de los valores atípicos o el escalado de los datos (introduciendo valores mínimo y máximo).

Red neuronal <https://www.neuraldesigner.com/learning/tutorials/neural-network>

Una red neuronal es un modelo computacional que se inspira en el funcionamiento del cerebro humano. Una arquitectura es una red neuronal con más de una neurona, con parámetros ajustables para cada neurona (pesos y sesgos).

Son de utilidad en un gran número de problemas que se pueden clasificar en tres tipos: aproximación (ajuste de una función a partir de los datos), clasificación (asignar un tipo a partir de unas características) o predicción, como en este trabajo.

Las neuronas se agrupan en capas, pudiendo estas ser de entrada, ocultas o de salida. Solamente hay una capa de entrada, que es la que recibe los datos, y una de salida, la que genera el modelo. Sin embargo, puede haber varias, una o ninguna capas ocultas. Existen varios tipos de capas.

La más usada universalmente es la capa de perceptrón (también conocido como capa densa). Como particularidad, la entrada se convierte en salida usando una función de activación, teniendo en cuenta unos pesos y sesgos. Existen distintas funciones de

activación, como la lineal, la tangente hiperbólica, o la función relu. Los pesos se asocian a una conexión entre dos neuronas y reflejan la intensidad de esta con un valor numérico w_{ij} , siendo i y j dos neuronas. Los sesgos son pesos que se definen para una neurona en concreto con la tarea de mejorar el aprendizaje. En una capa de perceptrón, lo primero que se aplica es una función de combinación (de las entradas con los pesos y sesgos), y después la función de activación para obtener la salida.

IMAGEN PERCEPTRON Y LSTM

Otros tipos de capa son la probabilística, usada principalmente en problemas de clasificación por lo que no se entrará en detalle, y la capa LSTM (long-short term memory). Estas son ampliamente usadas en problemas de predicción. Recibe la información en un conjunto de entradas y esta se procesa en distintas “puertas”, conocidas como la de olvido, la de entrada, la de estado y la de salida. La mayor diferencia viene dada porque guarda los estados intermedios en una celda, teniendo cierta memoria, que es lo que le da su nombre a la capa. Son bastante más complejas que la capa de perceptrón y no se pueden tratar de forma matricial como estas otras, pero presentan buenos resultados en conjuntos de datos con series temporales.

Por último, existen otro tipo de capas como las de escalado, desescalado y límite que sirven para tareas concretas como mantener los valores dentro de unos rangos o devolverlos a los rangos iniciales.

Es interesante mencionar que estas capas se pueden combinar, utilizando por ejemplo una capa oculta de perceptrón y otra LSTM. Encontrar la arquitectura adecuada para la red proporcionará mejores resultados.

Función de coste (error) <https://www.neuraldesigner.com/learning/tutorials/training-strategy>

El concepto de función de coste describe la tarea que la red neuronal va a realizar. Permite conocer una estimación sobre la calidad de los resultados obtenidos por un modelo. El objetivo siempre será minimizar esta función, lo que en nuestro caso significa mejores predicciones.

Esta función se compone de dos términos, el de error y el de regularización. El término de error es el más importante y un concepto fundamental en este trabajo. Mide el ajuste

de la red neuronal al conjunto de datos que se le ha suministrado. En un problema de predicción, indica la diferencia entre las predicciones que ha hecho la red neuronal y los resultados correctos.

Existen diferentes tipos de error, cada uno con ventajas y desventajas. IMÁGENES PARA CADA UNO, QUIZAS MEJOR SOLO DE LOS QUE USE EN EL APARTADO DE ASPECTOS RELEVANTES.

- La suma de los errores al cuadrado es uno de los más típicos y tiene la ventaja de que puede ser tratado como una función continua diferenciable.

- El error cuadrático medio es similar al anterior pero el error no aumenta con el tamaño del dataset, lo que es útil en conjunto de datos grandes.

- La raíz del error cuadrático medio es la raíz del error mencionado anteriormente.

- El error cuadrático normalizado tampoco aumenta con el tamaño del dataset e introduce un coeficiente de normalización con el que no contaban los anteriores.

- El error de Minkowski evita que el error esté muy marcado por unas pocas muestras con errores muy altos, que es un problema común a todos los anteriormente vistos.

Por otro lado, el término de regularización puede que exista o no, en contraposición al término de error. Un modelo es regular si pequeños cambios en las entradas suponen pequeños cambios en las salidas. Si no fuera así, es irregular por lo que un término de regularización se introduce para controlar la complejidad de la red neuronal. Internamente, hace que los pesos y sesgos sean más pequeños. Existen dos tipos principales:

- Regularización L1: utiliza la suma de los valores absolutos de los parámetros de la red neuronal.

- Regularización L2: utiliza la suma cuadrática de los parámetros de la red neuronal. Se aumenta o disminuye hasta encontrar un balance idóneo.

[https://www.neuraldesigner.com/blog/5 algorithms to train a neural network](https://www.neuraldesigner.com/blog/5%20algorithms%20to%20train%20a%20neural%20network)

Con el objetivo de minimizar la función de coste, es necesario optimizar la red neuronal. Este proceso trata de encontrar los parámetros (pesos y sesgos) óptimos, y para ello se utiliza un algoritmo de entrenamiento.

En primer lugar, las muestras del subconjunto de entrenamiento se introducen en la capa de entrada de la red, y se combinan con los pesos y sesgos de esta. Posteriormente se utiliza la función de activación de las neuronas. Acto seguido, se propagan las salidas de la primera capa a la siguiente, y así sucesivamente hasta llegar a la capa de salida. Por último, se calcula el gradiente del error mediante un algoritmo de retropropagación. Si este satisface las condiciones de finalización (que pueden ser un valor mínimo, un tiempo máximo, un número de iteraciones máximo o similares), termina el proceso, en cambio si no lo hace se modifican los parámetros y se repite todo el proceso. Cada repetición del proceso es lo que se conoce como un epoch.

Existen varios algoritmos de optimización, con diferentes características matemáticas y parámetros diferentes para además de minimizar el error, reducir el tiempo de entrenamiento y el número de iteraciones lo máximo posible. Entre los más famosos destacan el algoritmo de Levenberg-Marquardt, el método de Quasi-Newton y el algoritmo de estimación de momento adaptativo. Normalmente, el mejor algoritmo dependerá del tamaño del conjunto de datos. Para uno pequeño, Levenberg-Marquardt presenta buenos resultados por su rapidez de entrenamiento, aunque utiliza mayor cantidad de memoria que los demás, por eso no es bueno en tamaños mayores. En intermedios, Quasi-Newton trabaja mejor. Por último, en conjuntos de datos muy grandes el algoritmo de estimación del momento adaptativo, siendo el más moderno y adecuado a las nuevas necesidades, funciona mejor que el resto.

TECNICAS Y HERRAMIENTAS

Metodología

Necesitamos una metodología de desarrollo de software que se adapte a las características de este sistema. Debido a ello, metodologías tradicionales, que impliquen una planificación total del trabajo antes de comenzar el desarrollo, no serían óptimas para el problema en cuestión. En este sentido, metodologías ágiles que permitan un desarrollo iterativo e incremental, en vez de lineal, pueden ser de más utilidad para los objetivos de este trabajo.

De esta forma, un marco de trabajo ágil como el que podría ser Scrum, solapando fases del desarrollo (como en nuestro caso podrían ser el diseño de la interfaz web y las pruebas del modelo), con una estrategia de desarrollo incremental y una toma de decisiones a corto plazo, pueden proporcionar un método efectivo para conseguir mayor flexibilidad a cambios, mayor productividad y mejor calidad del software. Sin embargo, la particularidad de este sistema es su enfoque científico, no tan centrado en diseño de software clásico.

Por esta razón, la metodología que se va a usar en este proyecto es DSRM (Design Science Research Methodology), expuesta por Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger y Samir Chatterjee en <https://www.tandfonline.com/doi/abs/10.2753/MIS0742-1222240302>. Incorpora principios, prácticas y procedimientos necesarios para llevar a cabo una investigación científica y cumplir con tres objetivos principales: ser coherente con la literatura previa, proporcionar un modelo de proceso para hacer investigación en Design Science y proporcionar un modelo mental para presentar y evaluar la investigación de Design Science en un sistema de información como el de este trabajo.

Se incluyen seis pasos a seguir: la identificación del problema y motivación, definición de objetivos para una solución, diseño y desarrollo, demostración, evaluación y comunicación. IMAGEN PAGINA 11 PAPER 2007. Asimismo, el proceso permite un desarrollo iterativo acorde con las necesidades mencionadas previamente. Por lo tanto, aplicar esta metodología al sistema nos permitirá seguir un marco de trabajo y obtener una mayor calidad en el resultado final.

Motor

El motor de software usado para el diseño de la red neuronal y la obtención del modelo de predicción va a ser la biblioteca de código abierto OpenNN. Esta librería permite construir redes neuronales artificiales con un muy buen rendimiento, obteniendo mejores resultados en aspectos como la velocidad de ejecución y la asignación de memoria que otras bibliotecas de código abierto relacionadas con la inteligencia artificial como TensorFlow o PyTorch.

Haciendo uso de esta librería, el software Neural Designer, creado por antiguos alumnos de la Universidad de Salamanca, permite la creación de redes y modelos de manera intuitiva y rápida. Adopta algoritmos de entrenamiento e índices de error como los expuestos en el apartado (NUM CONCEPTOS TEORICOS), permitiendo poner en práctica los conceptos teóricos previamente mencionados.

Lenguajes programación

El modelo creado a partir de la red neuronal, además de la expresión matemática, se definirá en lenguaje Python, simplificando la sintaxis. De igual manera, podría traducirse y adaptarse a cualquier otro lenguaje de programación estructurada, al contar con capas de entrada y salida en un orden determinado.

Para la visualización de los resultados obtenidos a partir del modelo, usamos una interfaz web con Node.js como entorno de ejecución de código abierto y Express como su framework de aplicaciones, facilitando el desarrollo de aplicaciones web basadas en Node. Además, facilita la integración con bibliotecas de creación de gráficos (Charts.js) y de obtención de datos a través de APIs (Axios) que son necesarias para su puesta en marcha.

De esta manera, aprovechando la sencillez de los tres lenguajes básicos de creación de sitios web (HTML, CSS y JavaScript) y esta estructura, permitimos que personas sin conocimientos extensos sobre la inteligencia artificial puedan tener una forma gráfica de interactuar con el sistema, dotándolo de una capa de transparencia que oculta el modelo matemático y se centra en los datos finales.