

INFORME PROYECTO – FUNDAMENTOS DE DEEP LEARNING

PREDICCIÓN DE VARIABLES CLIMATICAS MAXPLANK






CONTEXTO

La predicción de variables climáticas, día a día a toma mayor relevancia a nivel global, esto gracias a que en diferentes ciudades se han venido implementando sistemas de identificación de variables climáticas, los cuales permiten la descripción ambiental enfocado principalmente en la mitigación de riesgos. Dado que día a día se generan un gran número de datos es necesario la implementación de modelos los cuales nos permitan su interpretación y aprovechamiento. Particularmente, un enfoque de esto es el aprovechamiento de las variables para la generación de modelos predictivos los cuales permitan la predicción de las variables climáticas en tiempos futuros.

Basados en esta necesidad, se aborda este enfoque desde el planteamiento de modelos basados en redes neuronales, y para evaluarlo se propone aprovechar el dataset publico Max Planck Weather, este conjunto de datos contiene 14 características diferentes, como la temperatura del aire, la presión atmosférica, la humedad, entre otros; los cuales fueron recopilados cada 10 minutos, a partir de 2003. Dado que existe diferentes metodologías de análisis de series de tiempo que busca la implementación de modelos predictivos, en este trabajo se propone la comparación de tres tipologías de modelo, en primer caso se abordara un enfoque clásico de análisis de series tiempo, mediante un modelo de base conocido como el modelo ingenuo o *naive model*, posteriormente se realiza una implementación de un red neural multicapa (Ambos modelos basado en: https://www.tensorflow.org/tutorials/structured_data/time_series), finalmente será implementado un modelo de aprendizaje evolutivo, los cuales han ganado interés en el análisis de series de tiempo (Modelo basado en: <https://towardsdatascience.com/unit-3-application-evolving-neural-network-for-time-series-analysis-63c057cb1595>). Los modelos implementados serán evaluados en función a su capacidad de predicción de las variables de interés.

DESCRIPCIÓN DE ESTRUCTURA DE NOTEBOOKS

Se crea un repositorio github con la siguiente estructura:

 01_exploracion_de_datos.ipynb	Created using Colaboratory
 02_Modelos_Weather_Time.ipynb	Created using Colaboratory
 ENTREGA1.pdf	Add files via upload
 INFORME PROYECTO.pdf	Add files via upload
 max_planck_weather_ts.csv	Create max_planck_weather_ts.csv

En el existe dos notebooks, el primero de ello llamado 01_exploracion_de_datos.ipynb, consiste en un análisis de los datos ambientales que contiene la base de datos max_planck_weather_ts.csv

(también en el repositorio), en este notebook mediante la herramienta ProfileReport de la librería pandas se analizan las características de cada variable y la correlación entre ellas.

El segundo notebook tiene como nombre 02_Modelos_Weather_Time.ipynb consiste en el análisis de los modelos propuestos en función al dataset en cuestión. Este notebook cuenta con la estructuración de tres modelos: modelo naïve, modelo red neuronal multicapa y modelo de aprendizaje evolutivo, los dos primeros de ellos son ejecutados usando Keras. A continuación se realiza una descripción de la metodología utilizada.

DESCRIPCIÓN DE TU SOLUCIÓN

Como primera parte en el proceso de modelamiento se definen las particiones a utilizar para las etapas de entrenamiento, testeo y validación de modelos. En series de tiempo estas particiones deben hacerse de manera secuencial, el Split propuesto para el análisis a realizar es:

- 70% inicial de datos para entrenamiento.
- 20% siguiente de datos para validación.
- 10% final de datos para testeo.

De esta manera se garantiza mantener la secuencia y predictibilidad de los datos, además se garantiza que las etapas de testeo y validación sean realizadas con datos que no haya sido vistos previamente por el modelo. Además, cabe anotar que se realiza una normalización de los datos (restando su media y dividiéndolos en su desviación estándar) para poder alimentarlos a la red neuronal. Adicionalmente, se realiza un re – muestreo de los datos en intervalos de 60 minutos con la finalidad de disminuir la cantidad de información para procesamiento, pero sin perder información relevante para la predicción de la serie de tiempo. Visualmente, con una ventana de tiempo corta y con tres muestras aleatorias los pronósticos ingenuos de la variable T (degC) se pueden apreciar en la figura 1.

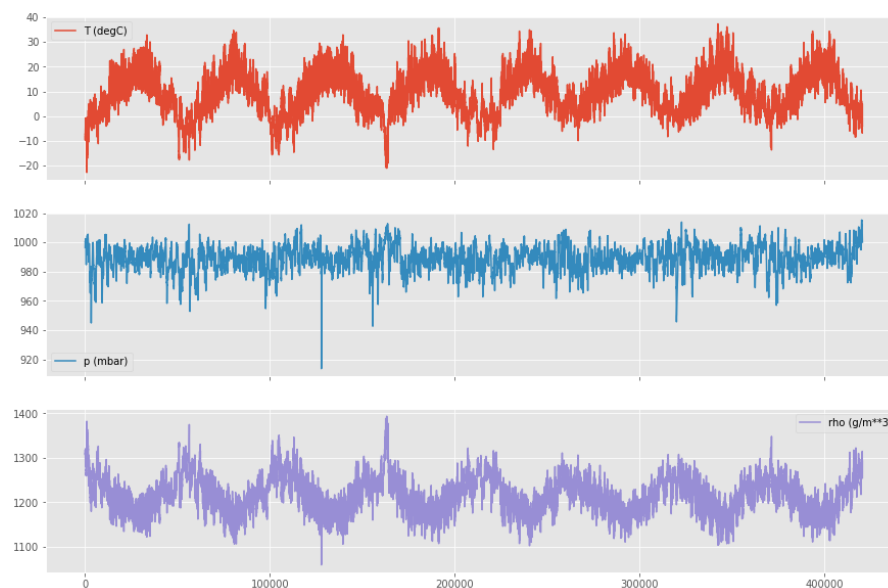


Figura 1. Re – muestreo de datos para su implementación en una red neuronal

Como se fue mencionado en el contexto del trabajo, buscamos realizar la comparación de tres tipologías de modelos para el análisis de series de tiempo, los modelos propuestos se describen a continuación:

Para la predicción de series de tiempo suele utilizarse un modelo de base conocido como el *modelo ingenuo* o *naive model*, este corresponde a realizar la predicción del instante t igualándola al valor de la variable en el tiempo $t - 1$. Esta predicción sirve como punto de partida y suele utilizarse para comparar los modelos posteriores.

Por otro lado, también se propone un modelo el cual busca realizar la predicción de datos a partir de una red neuronal. El modelamiento propuesto como base de análisis es una red neuronal densa que toma inputs y genera los resultados con una estructura similar a la apreciable en la figura 2.

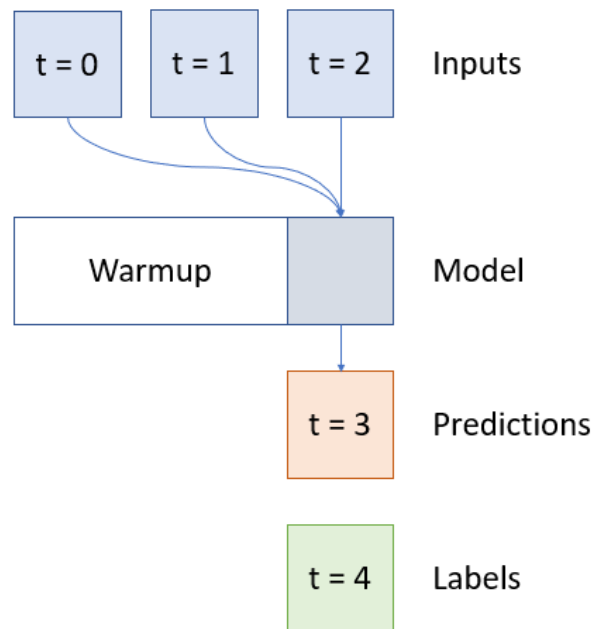


Figura 2 Estructura de modelo propuesto (Tomado de:
https://www.tensorflow.org/tutorials/structured_data/time_series)

En esta estructura el modelo toma los valores de la variable a predecir en tiempos previos (para este caso en $t = 0$, $t = 1$, $t = 2$) y con estos alimenta la red para poder generar una predicción de valor en $t = 3$. Bajo esta estructura y con una ventana de 24 datos de tiempo (1 día) se realiza la predicción de datos para la comparación de las distintas variables.

Finalmente, el algoritmo evolutivo propuesto sigue los siguientes pasos:

- Definición del genotipo: Para considerar el algoritmo propuesto como uno evolutivo se debe definir el genotipo (características intrínsecas) de los individuos. En este caso, consideraremos cada individuo como una red neuronal que sería posteriormente combinada para la generación de nuevos individuos. Los diferentes “genes” de los organismos serán los pesos y bias de la red. En este sentido el genotipo será la matriz de pesos/bias y su tamaño dependerá de la estructura definida; basándonos en la documentación inicial la estructura propuesta tendrá una estructura 5/5/5 (densidad de las

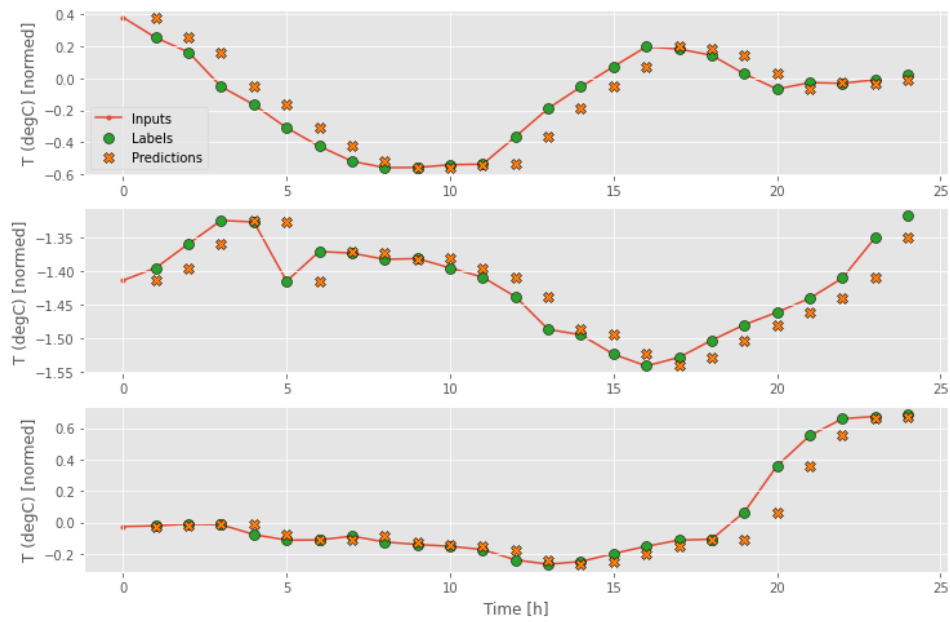
capas internas). La generación de los primeros individuos del algoritmo (primera generación) será a partir de una distribución normal aleatoria entre de -1 a 1 por facilidad, además la función de activación ReLU.

- Definición de cruces: Los cruces entre los padres corresponden a la manera en que los genotipos (matrices) interactúan y como posteriormente dan paso a la generación de nuevos individuos. Para este caso el cruce se realizará a partir de una combinación lineal de los padres con una constante de cruce (constaros) que se usa como operador lineal para sesgar la matriz de los hijos hacia uno de sus padres y este valor será un número aleatorio con media 5 y desviación estándar 0.15. En el algoritmo propuesto se generarán 4 hijos en cada cruce de la siguiente manera:
 - Hijo 1: Cruce lineal de p1 y p2 con una constante de cruce aleatoria c1.
 - Hijo 2: Cruce lineal de p1 y p2 con una constante de cruce aleatoria c2.
 - Hijo 3: Cruce lineal de p1 y p2 con una constante de cruce aleatoria c3, con una posterior mutación de todos los pesos bias de la matriz por la constante de mutación m3.
 - Hijo 4: Cruce lineal de p1 y p2 con una constante de cruce aleatoria c4, con una posterior mutación de todos los pesos bias de la matriz por la constante de mutación m4.
- Selección de mejor individuo: Se realiza una comparación de los 6 individuos de cada cruce (2 padres y 4 hijos) y se selecciona el mejor de los 6 para avanzar a la siguiente generación. La función de fitness definida para la selección de los individuos será el MSE entre la predicción y los valores reales de las series de tiempo. Se propone un máximo de 20 iteraciones para el desarrollo del algoritmo evolutivo. Esta estrategia de predicción será aplicada a las tres variables seleccionadas previamente la temperatura en grados Celsius, la densidad del aire en g/m3 y la presión en milibares

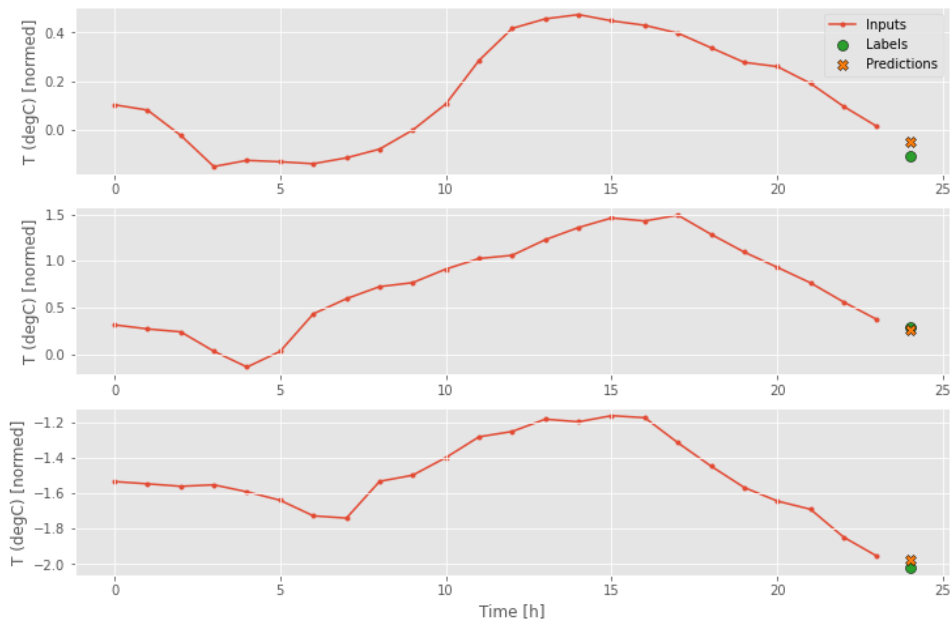
DESCRIPCIÓN DE LOS RESULTADOS

Podemos observar el comportamiento predictivo similar en los modelos propuesto, a continuación, se muestra gráficamente el comportamiento de los tres modelos. Se evidencia solo para la variable temperatura, el comportamiento de las demás variables puede visualizarse en el notebook.

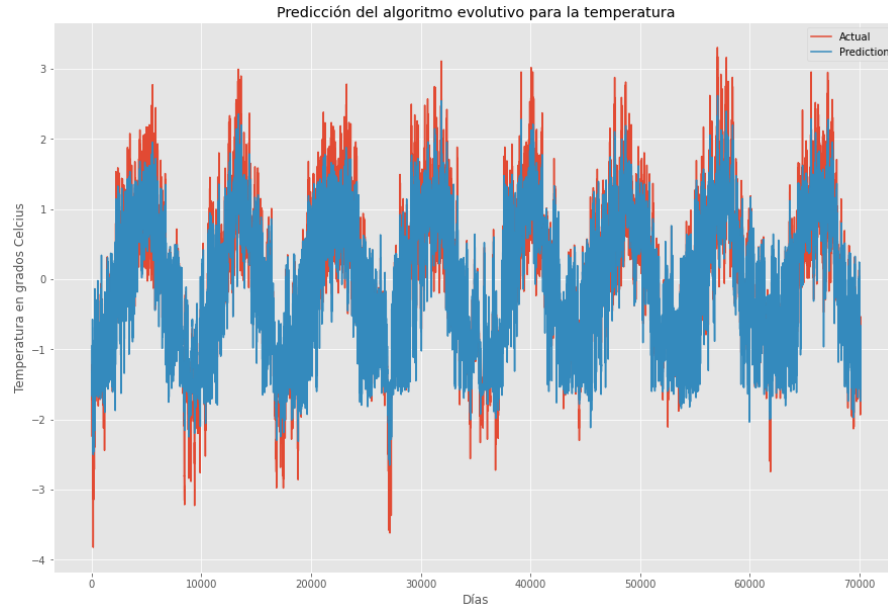
Mediante la metodología de linebase propuesto en el modelo naive es posible evidenciar el siguiente comportamiento predictivo:



Para la red neuronal propuesta, es posible visualizar un comportamiento predictivo similar:



Finalmente, para el caso del algoritmo evolutivo podemos observar que la predicción la serie de tiempo de forma adecuada, para de la variable temperatura se muestra a continuación:



Se determina el error medio cuadrático (MSE) para cada modelo propuesto, con la finalidad de determinar cual de ellos cuenta con una mejor predicción de las variables climáticas.

<i>Variable</i>	<i>MSE modelo naive</i>	<i>MSE Red Neuronal</i>	<i>MSE modelo evolutivo</i>
<i>T (degC)</i>	0.0852	0.0686	0.0978
<i>p (mbar)</i>	1.1652	0.0514	0.0494
<i>rho (g/m**3)</i>	1.588	0.0568	0.1807

Es posible evidenciar inicialmente que la red neuronal propuesta cuenta con un error medio cuadrático mejor para todas las variables climáticas que el modelo naive, indicando que puede describir mejor las variables de interés. Por otro lado, el modelo evolutivo propuesto no logra disminuir significativamente el error medio cuadrático con respecto a la red neuronal propuesta, de hecho en algunas variables climáticas incrementa significativamente el error en la predicción.

Conclusiones

- Si bien los algoritmos evolutivos son atractivos por la capacidad de hallar soluciones disruptivas y diferenciadas a otras aproximaciones, el tiempo que puede tomar implementarlas y el tiempo que requieren para ser ejecutadas puede ser muy elevada. En el caso evaluado en particular tanto los tiempos de procesamiento como los resultados de las redes neuronales son mejores para la red neuronal implementada con la librería keras frente a los demás modelos implementados.
- A pesar de que el enfoque de modelos naive son frecuentemente utilizados en el análisis de series de tiempos y cuentan con una rápida implementación, los errores en la predicción siempre son mayores en comparación con los modelos de redes neuronales multicapa.
- El modelo de red neuronal propuesta permite la predicción de las variables climáticas propuestas con un error medio cuadrático adecuado. Este tipo de herramientas pueden ser extrapolar a otros problemas en la predicción de variables climáticas.