

Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

Grado en Ingeniería Informática
(Mención de Computación)

**Implementación de algoritmos de
aprendizaje automático para la
predicción del consumo eléctrico en el
sector energético**

Autor:
D.Pablo Bueno Sánchez

Índice

Agradecimientos	4
Resumen	5
Abstract	6
Índice de Figuras	6
Índice de Tablas	7
Índice de Ecuaciones	8
1. Introducción	8
1.1. Contexto y Motivación	8
1.2. Objetivos	8
1.3. Organización y Estructura de la memoria	8
2. Estado del Arte	8
2.1. Introducción	8
2.2. Inteligencia Artificial	8
2.3. Machine Learning	8
2.3.1. aprendizaje Supervisado	8
2.3.2. Regresión Lineal	9
2.3.3. Árboles de Decisión	9
2.3.4. XGBoost	9
2.3.5. Otros Modelos	9
2.4. Deep Learning	9

2.4.1.	Redes Neuronales Artificiales	9
2.4.2.	Redes Neuronales Recurrentes (RNN)	9
2.4.3.	Backpropagation Through Time y problemas de evanescencia / explosión del gradiente	10
2.4.4.	Redes Long Short-Term Memory (LSTM)	11
2.4.5.	Gated Recurrent Unit (GRU)	12
2.4.6.	Redes Neuronales Convolucionales (CNN)	12
2.4.7.	Autoregressive Integrated Moving Average (ARIMA)	12
2.5.	Series Temporales	12
2.5.1.	Tendencia	13
2.5.2.	Estacionalidad	13
2.5.3.	Residual	13
2.5.4.	Series Estacionarias	13
2.5.5.	Predicción	13
3.	Datos	13
3.1.	Obtención	13
3.2.	Descripción	13
4.	Preprocesamiento	15
4.1.	Obtención de características y unificación de los datos	15
4.1.1.	train.csv	16
5.	Predicción del consumo energético	17
5.1.	Base teórica	17
5.1.1.	Algoritmos	17

5.1.2. Funciones de pérdida	17
5.1.3. Optimizadores	17
5.1.4. Modelo 1	17
5.1.5. Modelo 2	17
5.1.6. Modelo 3	17
6. Predicción de producción energética	17
6.1. Base teórica	17
6.1.1. Algoritmos	17
6.1.2. Funciones de pérdida	17
6.1.3. Optimizadores	17
6.2. Implementación	17
6.2.1. Preprocesamiento de Datos	17
6.2.2. Clasificación Modelo 1	17
6.2.3. Clasificación Modelo 2	17
6.2.4. Clasificación Modelo 3	17
7. Conclusiones	18
7.1. Predicción de consumo energético	18
7.2. Predicción de producción energética	18
7.3. Comparación de resultados	18
7.4. Aplicabilidad de los algoritmos de predicción	18
8. Trabajo futuro	18
Bibliografía	18

Agradecimientos

Resumen

Abstract

Índice de Figuras

Índice de Tablas

Índice de Ecuaciones

1. Introducción

1.1. Contexto y Motivación

1.2. Objetivos

1.3. Organización y Estructura de la memoria

2. Estado del Arte

2.1. Introducción

2.2. Inteligencia Artificial

2.3. Machine Learning

Dentro de la computación, los algoritmos son los encargados de regir el comportamiento de las máquinas para la resolución de problemas. Estos algoritmos reciben una entrada y, a partir de una serie de pasos, producen una salida deseada para un problema en concreto. Sin embargo, existen problemas para los cuales no se tiene ningún algoritmo, aquí es donde es posible hacer uso del machine learning.

El Machine Learning es una rama dentro de la Inteligencia Artificial la cual se encarga de permitir a las máquinas extraer, a partir de una gran cantidad de datos, una aproximación fiable de algoritmos para la resolución de diversos problemas.

2.3.1. aprendizaje Supervisado

El aprendizaje supervisado es una categoría del machine learning caracterizada por el uso de datos etiquetados para el entrenamiento de los modelos.

El uso de conjuntos de datos etiquetados permite a los modelos modificar gradualmente su comportamiento para poder ajustarse al modelo final.

Con el fin de poder modificar los componentes del modelo se utilizan funciones de pérdida,

las cuales permiten calcular la precisión de la salida del modelo con respecto a la salida deseada, permitiendo así minimizar el error hasta un mínimo deseable.

2.3.2. Regresión Lineal

2.3.3. Árboles de Decisión

2.3.4. XGBoost

2.3.5. Otros Modelos

2.4. Deep Learning

2.4.1. Redes Neuronales Artificiales

2.4.2. Redes Neuronales Recurrentes (RNN)

Las redes neuronales recurrentes o RNN son un tipo de redes neuronales artificiales diseñadas principalmente para la detección de patrones en secuencias de datos [1].

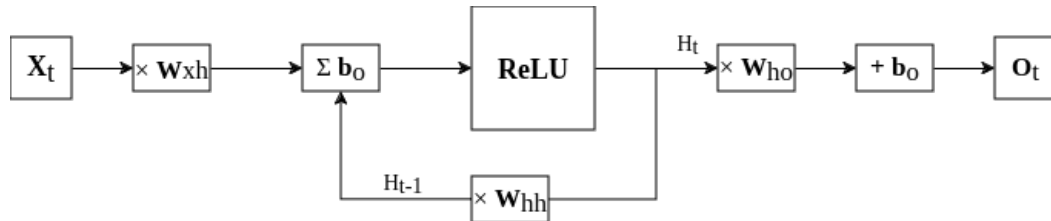


Figura 1: Estructura de una red RNN con función de activación ReLU

Al igual que otros tipos de redes neuronales, como los perceptrones multicapa (MLP), la información se transmite desde la capa de entrada \mathbf{X}_t hasta la capa de salida \mathbf{O}_t pasando por el bloque de capa oculta \mathbf{H}_t . Sin embargo, a diferencia de estas, la entrada del bloque de la capa oculta depende tanto de la entrada \mathbf{X}_t como de la salida del bloque en su iteración anterior \mathbf{H}_{t-1} . Ver figura 1.

La salida de los bloques \mathbf{H}_t y \mathbf{O}_t viene dada por las fórmulas:

$$\mathbf{H}_t = \phi_h(\mathbf{X}_t \mathbf{W}_{xh} + \mathbf{H}_{t-1} \mathbf{W}_{hh} + \mathbf{b}_h) \quad (1)$$

$$\mathbf{O}_t = \phi_o(\mathbf{H}_t \mathbf{W}_{ho} + \mathbf{b}_o) \quad (2)$$

Donde, \mathbf{X}_t es la salida de la capa de entrada, \mathbf{H}_t y \mathbf{H}_{t-1} son las salidas de la capa oculta en los instantes t y $t-1$, \mathbf{W}_{xh} , \mathbf{W}_{hh} , y \mathbf{W}_{ho} son las matrices de pesos entre las diferentes capas

de la neurona y \mathbf{b}_h y \mathbf{b}_o son los vectores de bias o sesgo para la capa oculta y la de salida respectivamente.

ϕ_h y ϕ_o son las funciones de activación de las capas oculta y de salida respectivamente. En el caso de ϕ_h suele ser la función rectificador (ReLU) o tangente hiperbólica (tanh).

2.4.3. Backpropagation Through Time y problemas de evanescencia / explosión del gradiente

Para poder comprender el principal problema de las redes neuronales recurrentes es necesario conocer el funcionamiento del algoritmo de retropropagación a través del tiempo o BPTT (Back-Propagation Through Time).

Con el fin de entrenar una RNN, primero se debe establecer una función de pérdida \mathcal{L} , la cual nos permite conocer la distancia entre el valor de salida de la neurona \mathbf{O}_t y el valor real \mathbf{Y}_t .

$$\mathcal{L}(\mathbf{O}, \mathbf{Y}) = \sum_{t=1}^T \ell_t(\mathbf{O}_t, \mathbf{Y}_t) \quad (3)$$

Con el fin de minimizar dicha distancia, se han de calcular las derivadas parciales de \mathcal{L} con respecto los pesos de la neurona, \mathbf{W}_{xh} , \mathbf{W}_{hh} , y \mathbf{W}_{ho} . A partir de la ecuación 2 aplicando la regla de la cadena, se puede obtener:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{ho}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \frac{\partial \phi_o}{\partial \mathbf{W}_{ho}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{H}_t \quad (4)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \frac{\partial \phi_o}{\partial \mathbf{H}_t} \cdot \frac{\partial \mathbf{H}_t}{\partial \phi_h} \cdot \frac{\partial \phi_h}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \cdot \frac{\partial \mathbf{H}_t}{\partial \phi_h} \cdot \frac{\phi_h}{\partial \mathbf{W}_{hh}} \quad (5)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \frac{\partial \phi_o}{\partial \mathbf{H}_t} \cdot \frac{\partial \mathbf{H}_t}{\partial \phi_h} \cdot \frac{\partial \phi_h}{\partial \mathbf{W}_{xh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \cdot \frac{\partial \mathbf{H}_t}{\partial \phi_h} \cdot \frac{\phi_h}{\partial \mathbf{W}_{xh}} \quad (6)$$

Las ecuaciones 5 y 6, al depender \mathbf{H}_t de sus valores anteriores, pueden reescribirse como:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \sum_{k=1}^t \frac{\partial \mathbf{H}_t}{\partial \mathbf{H}_k} \cdot \frac{\partial \mathbf{H}_k}{\partial \mathbf{W}_{hh}} \quad (7)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \sum_{k=1}^t \frac{\partial \mathbf{H}_t}{\partial \mathbf{H}_k} \cdot \frac{\partial \mathbf{H}_k}{\partial \mathbf{W}_{xh}} \quad (8)$$

Para entender el problema de la evanescencia y explosión del gradiente en las RNN es necesario fijarse en la expresión $\frac{\partial \mathbf{H}_t}{\partial \mathbf{H}_k}$ de las ecuaciones 7 y 8.

El problema de la evanescencia del gradiente se produce cuando, en secuencias largas de datos, se establece unos pesos iniciales menores a 1. A medida que se entrena la red RNN. Los gradientes se hacen menores con cada iteración hasta llegar a desaparecer, provocando que la red no pueda seguir entrenandose.

Con el problema de la explosión del gradiente se da una situación parecida a la evanescencia del gradiente, pero al contrario que esta, al establecer los pesos iniciales a un valor superior a 1, los gradientes se irán haciendo mayores con cada iteración hasta que sean tan grandes que sea imposible entrenar la red debido a que el valor de los pesos sufren grandes cambios en cada actualización, impidiendo encontrar un resultado óptimo.

2.4.4. Redes Long Short-Term Memory (LSTM)

Con el fin de solventar los problemas de la evanescencia y explosión del gradiente de las redes recurrentes, en 1997 se ideó una nueva estructura de red recurrente.

La principal diferencia entre una red LSTM y una RNN es la presencia de un nuevo tipo de memoria a largo plazo la cual ayuda a regular la memoria a corto plazo y predecir la salida de dicha red.

La estructura principal se caracteriza por su división en 3 puertas principales.

La primera de las puertas se denomina “puerta del olvido” (\mathbf{F}_t). Es la parte de la red que se encarga de decidir que porcentaje de la memoria a largo plazo (\mathbf{C}_{t-1}) se debe olvidar o mantener.

La segunda puerta es denominada la “puerta de entrada” (\mathbf{I}_t), determina la cantidad de memoria potencial a largo plazo debe añadirse.

Por último se encuentra la “puerta de salida” (\mathbf{O}_t) la cual se encarga de determinar cuánta memoria a corto plazo debe recordarse.

$$\mathbf{F}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xf} + \mathbf{H}_{t-1} \mathbf{W}_{hf} + \mathbf{b}_f) \quad (9)$$

$$\mathbf{I}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xi} + \mathbf{H}_{t-1} \mathbf{W}_{hi} + \mathbf{b}_i) \quad (10)$$

$$\mathbf{O}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xo} + \mathbf{H}_{t-1} \mathbf{W}_{ho} + \mathbf{b}_o) \quad (11)$$

En las ecuaciones 9, 10 y 11 puede observarse que el funcionamiento de las puertas es idéntico salvo por los pesos \mathbf{W}_{lj} y los bias \mathbf{b}_k donde $l, j, k \in x, f, i, o, h$ entre la entrada, la capa oculta y las diferentes puertas.

**Estructura LSTM

Figura 2: Estructura de una red LSTM

Con el fin de actualizar la memoria a largo plazo, primero es necesario calcular la memoria potencial $\tilde{\mathbf{C}}$, la cual se calcula de manera similar al resto de puertas salvo por la función de activación, la cual en lugar de ser la función sigmoide σ utiliza una función de tangente hiperbólica \tanh para calcular su salida. Vease la ecuación 12.

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xc} + \mathbf{H}_{t-1} \mathbf{W}_{hc} + \mathbf{b}_c) \quad (12)$$

Con todas las puertas definidas, el valor de la memoria a largo plazo viene dado por la ecuación 13

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t \quad (13)$$

Por último, con la memoria a largo plazo actualizada, tanto el valor de la salida como el de la capa oculta actualizada, vendrá dado por la función ??

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t) \quad (14)$$

Pese a solentar los problemas de evanescencia del gradiente, es posible que pueda producirse explosión del gradiente si no se predefinen bien las componentes. Pese a ello las redes LSTM son, actualmente, uno de los modelos de deep learning con mejores resultados para datos secuenciales como series temporales o reconocimiento del language.

2.4.5. Gated Recurrent Unit (GRU)

2.4.6. Redes Neuronales Convolucionales (CNN)

2.4.7. Autoregressive Integrated Moving Average (ARIMA)

2.5. Series Temporales

Una serie temporal es una secuencia de datos obtenidos a lo largo de diferentes momentos ordenados conológicamente. Dichos datos pueden obtenerse y ordenarse de dos formas diferentes:

- **Equidistante en el tiempo:** Los datos se obtienen en intervalos iguales de tiempo, ya sea de forma diaria, mensual, anual u horaria.
- **No equidistante en el tiempo:** Los datos se obtienen en intervalos desiguales de tiempo, como por ejemplo la medición de los kilómetros recorridos por un vehículo cada vez que se realiza una revisión en el taller.

Una de las cualidades intrínsecas de las series temporales es que los datos adyacentes suelen tener una dependencia entre ellos. Dicha dependencia entre observaciones provee de un gran interés práctico. Una de las áreas de aplicación de dicha dependencia es la de predicción de futuros valores dentro de la serie.

2.5.1. Tendencia

2.5.2. Estacionalidad

2.5.3. Residual

2.5.4. Series Estacionarias

2.5.5. Predicción

asdasd

3. Datos

En este apartado se tratará de describir tanto la fuente como una descripción detallada de los datos que se han utilizado para la realización de las diferentes pruebas que se han realizado para el trabajo actual.

3.1. Obtención

El trabajo se ha realizado sobre un dataset obtenido de una competición de Kaggle llamada “*Enefit - Predict Energy Behavior of Prosumers*” realizada entre el 1 de Noviembre de 2023 y el 31 de Enero de 2024.

Para ayudar con el pre-procesamiento de los datos, se ha usado el dataset “*fabiendaniels-mapping-locations-and-county-codes*”, el cual ayuda a asignar un condado a una longitud y latitud específicas.

3.2. Descripción

El conjunto de datos consta de varios archivos csv con diversas variables tanto energéticas como meteorológicas. Dichos archivos fueron obtenidos para predecir tanto la producción como

el consumo de energía de diversas personas en Estonia con instalación de paneles solares en sus hogares.

A continuación se encuentra una descripción de cada archivo y todas sus variables:

1. **train.csv:**

- **county:** ID del condado.
- **is_business:** Indica si el consumidor es una empresa o un individuo.
- **product_type:** ID de un mapa de tipo de contrato.
- **target:** cantidad de consumo o producción del segmento relevante durante la hora.
- **is_consumption:** Indica si el elemento de la columna 'target' es de consumo o producción.
- **datetime:** Hora local de Estonia en EET (UTC+2)/ EEST(UTC+3). Define el inicio del periodo de una hora en el cual se da el valor de 'target'.
- **data_block_id:** ID relacionado con la fecha de la toma de los datos.
- **row_id:** Identificador único de la fila
- **prediction_unit_id:** Identificador único formado por la unión de las columnas 'country', 'is_business' y 'product_type'.

2. **gas_prices.csv:**

- **origin_date:** Fecha en el que los precios diarios se hicieron disponibles.
- **forecast_date:** Fecha en la que los precios previstos deberían ser relevantes
- **[lowest/highest]_price_per_mwh:** El precio mínimo/máximo del gas natural en el mercado diario ese día de negociación.
- **data_block_id** Identificador asociado a la fecha de registro de los datos.

3. **client.csv:**

- **product_type:** ID de un mapa de tipo de contrato.
- **country:** ID del código del país.
- **eic_count:** Número agregado de puntos de consumo (EICs - European Identifier Code).
- **installed_capacity:** Capacidad, en KWh, de los paneles solares fotovoltaicos instalados.
- **is_business:** Indicador booleano del tipo de cliente. (empresa o particular)
- **date:** Fecha de registro de los datos del cliente.
- **data_block_id:** Identificador asociado a la fecha de registro de los datos

4. **electricity_prices.csv:**

- **origin_date:** Fecha de la realización de la predicción.

- **forecast_date:** Fecha de la predicción de los datos.
- **euros_per_mwh:** Precio de la electricidad en MWh.
- **data.block_id:** Identificador asociado a la fecha de registro de los datos.

5. **historical_weather.csv:**

- **datetime:** Marca de tiempo de la toma de los datos.
- **temperature:** Temperatura ($^{\circ}\text{C}$) tomada al final de la medición.
- **dewpoint:** Punto de rocío ($^{\circ}\text{C}$).
- **rain:** Lluvia precipitada durante la marca de tiempo (mm^3)
- **snowfall:** Nieve precipitada durante la marca de tiempo (cm^3)
- **surface_pressure:** Presión del aire a nivel del suelo (hPa)
- **cloudcover_[low/mid/high/total]:** Nivel de cubrimiento del cielo por las nubes a alturas de 0-3, 3-8, 8+ km y el total.
- **windspeed_10m:** Velocidad del viento (m/s) a 10 metros sobre el nivel del suelo.
- **winddirection_10m:** dirección del viento (grados) a 10 metros sobre el nivel del suelo.
- **shortwave_radiation:** Radiación de onda corta (Wh/m^2)
- **direct_solar_radiation:** Radiación solar directa ($\frac{\text{W}}{\text{m}^3}$)
- **diffuse_radiation:** Radiación solar difusa ($\frac{\text{W}}{\text{m}^3}$)
- **[latitude/longitude]:** Coordenadas geográficas de la estación climática.
- **data.block_id:** Identificador asociado a la fecha de registro de los datos.

4. Preprocesamiento

El entrenamiento de los diferentes modelos de inteligencia artificial requiere de un correcto tratamiento de los datos de entrada para poder no solo ser entrenados, si no hacerlo de la manera mas óptima posible.

Los datos obtenidos desde la plataforma de kaggle han de pasar por un previo de procesamiento adecuado con el fin de adecuarlo lo mas posible a los modelos que serán entrenados con posterioridad.

El procesamiento de los datos estará dividido en diferentes partes claramente diferenciadas, las cuales serán explicadas en los siguientes apartados.

4.1. Obtención de características y unificación de los datos

El primer paso a tomar en el procesamiento de los datos será el de la unificación de los diferentes conjuntos de datos en un único dataset.

Para ello antes es necesario tratar dichos datos y obtener una serie de características únicas para poder obtener datos relevantes y un punto de conexión sobre el que podrán unirse mas adelante.

A continuación se listan los diferentes tratamientos por los que han pasado los diferentes datos.

4.1.1. `train.csv`

En el entrenamiento de series temporales, los modelos de aprendizaje automático no pueden procesar directamente los valores de tipo `datetime`, por lo que el principal tratamiento para el conjunto será el de aplicar la extracción de características temporales.

La extracción de características temporales o ingeniería de características temporales, como su nombre indica, consiste en transformar los datos de una variable temporal en variables numéricas o categóricas, las cuales permiten la identificación de patrones estacionales, tendencias y otros efectos que pueden influir en las series analizadas.

Las características obtenidas para el conjunto han sido las siguientes:

- **Date:** fecha compuesta únicamente por el año, mes y día, ignorando las marcas horarias (%YYYY- %MM- %DD).
- **Year:** Año de la fecha (%YYYY).
- **Quarter:** Cuatrimestre. Útil para la parte financiera, como los precios de consumo de la energía (1-3).
- **Month:** Mes del año (1-12).
- **Week:** Semana del año (1-55).
- **Hour:** Hora del día (0-23).
- **Day of year:** Día del año (1-355:366)
- **Day of month:** Día del mes (1-27:28:30:31)
- **Day of week:** Día de la semana (1-7)

5. Predicción del consumo energético

5.1. Base teórica

5.1.1. Algoritmos

5.1.2. Funciones de pérdida

5.1.3. Optimizadores

5.1.4. Modelo 1

5.1.5. Modelo 2

5.1.6. Modelo 3

6. Predicción de producción energética

6.1. Base teórica

6.1.1. Algoritmos

6.1.2. Funciones de pérdida

6.1.3. Optimizadores

6.2. Implementación

6.2.1. Preprocesamiento de Datos

6.2.2. Clasificación Modelo 1

6.2.3. Clasificación Modelo 2

6.2.4. Clasificación Modelo 3

asdasd w fqf

7. Conclusiones

asdjsadjajsd

7.1. Predicción de consumo energético

akdjaskdjkasj

7.2. Predicción de producción energética

7.3. Comparación de resultados

7.4. Aplicabilidad de los algoritmos de predicción

8. Trabajo futuro

Bibliografía

Referencias

- [1] Robin M. Schmidt. «Recurrent Neural Networks (RNNs): A gentle Introduction and Overview». Tesis de mtría. Eberhard-Karls-University Tübingen, 2019.