



UNIVERSIDAD
DE GRANADA

Escuela Técnica Superior de Ingenierías Informática y
Telecomunicación

MÁSTER UNIVERSITARIO EN CIENCIA DE DATOS E INGENIERÍA
DE COMPUTADORES

TRABAJO DE FIN DE MÁSTER

Estudio sobre la Efectividad del Positional Encoding en Transformers para Series Temporales y Diseño de Mecanismos Adaptados

Presentado por:
Cristhian Moya Mota

Curso académico 2024-2025



Estudio sobre la Efectividad del Positional Encoding en Transformers para Series Temporales y Diseño de Mecanismos Adaptados

Cristhian Moya Mota

Cristhian Moya Mota *Estudio sobre la Efectividad del Positional Encoding en Transformers para Series Temporales y Diseño de Mecanismos Adaptados.*
Trabajo de fin de Grado. Curso académico 2024-2025.

**Responsable de
tutorización**

Julian Luengo Martín
DECSAI
Diego Jesús García Gil
LSI

Máster Universitario en
Ciencia de Datos e
Ingeniería de
Computadores
Escuela Técnica Superior
de Ingenierías Informática
y Telecomunicación
Universidad de Granada

DECLARACIÓN DE ORIGINALIDAD

D./Dña. Cristhian Moya Mota

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2024-2025, es original, entendido esto en el sentido de que no he utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 21 de julio de 2025

Fdo: Cristhian Moya Mota

Dedicatoria (opcional)

Ver archivo preliminares/dedicatoria.tex

Agradecimientos

Agradecimientos (opcional, ver archivo preliminares/agradecimiento.tex).

Summary

An english summary of the project (around 800 and 1500 words are recommended).

File: preliminares/summary.tex

Índice general

Agradecimientos	V
Summary	VII
1. Introducción	1
1.1. Motivación	1
1.1.1. Importancia de las Series Temporales	1
1.1.2. La dificultad de predicción a largo plazo: información posicional	3
1.2. Justificación	4
1.3. Objetivos	4
2. Tendencias y Estado del arte	7
2.1. Procesamiento básico de Series Temporales	7
2.1.1. Temporalidad: estructura y frecuencia de muestreo	8
2.1.2. Valores faltantes	8
2.1.3. Normalización de los datos	8
2.1.4. Métodos para partición y evaluación del rendimiento	8
2.2. Modelos basados en descomposición	8
2.3. Transformers y Positional Encoding	8
2.4. Conjuntos de datos disponibles	8
3. Selección y preprocesado de los conjuntos de datos	9
4. Modelos de encoding posicional y entorno de trabajo	11
5. Análisis comparativo de positional encoding sobre las bases de datos	13
6. Conclusiones y trabajos futuros	15
A. Ejemplo de apéndice	17
Glosario	19
Bibliografía	21

1. Introducción

En este primer punto, describiremos de forma breve la motivación a realizar este TFM. Analizaremos la importancia de poseer herramientas capaces de realizar una predicción a largo plazo en series temporales (en inglés, *Long-term Time Series Forecasting*, LSTF), pero, sobre todo, centrándonos en un aspecto clave: la codificación posicional (*Positional Encoding*, en adelante, PE), común a prácticamente todas las alternativas actuales del estado del arte.

Procederemos a justificar su importancia, así como a formular los objetivos que se cubren con la realización de este trabajo.

1.1. Motivación

En la actualidad, los datos son uno de los bienes más preciados. Las telecomunicaciones nos han permitido alcanzar un volumen inimaginable de información digital, la cual no somos prácticamente capaces de procesar, y extraer conocimiento útil se convierte en una tarea complicada. Su gran variedad y modalidad hace necesario disponer de modelos multimodales cada vez más complejos para procesarlos, como los modelos fundacionales [1], para tratar así de disponer de una herramienta cercana a ser capaz de procesar todo tipo de información.

Hemos podido apreciar grandes avances en el procesamiento de texto, con los grandes modelos de lenguaje como GPT, el cual se ha convertido en una herramienta que usamos habitualmente para resolver nuestras dudas. Ahora, incluso permite generar imágenes, video y audio, y pensar profundamente las respuestas.

Sin embargo, estos no son los únicos tipos de datos que podemos emplear para aprender. Los datos basados en flujos, y las series temporales, son un recurso clave que podemos emplear para resolver multitud de problemáticas. Podemos predecir a largo plazo, clasificar fenómenos, o bien, incluso detectar anomalías. En este trabajo, nos centraremos sobre todo en la primera tarea y las dificultades que existen en este ámbito.

1.1.1. Importancia de las Series Temporales

Las series temporales son un tipo de datos caracterizados por una secuencia organizada en el tiempo. Generalmente, consisten en una o varias variables observadas a intervalos regulares, con el objetivo de registrar la evolución de un fenómeno. Suelen recogerse de forma periódica mediante sensores o procesos automatizados (por ejemplo, la temperatura de un motor medida por un termostato electrónico), aunque también pueden originarse de forma irregular, como en el caso de registros manuales. Es importante que, cuando se trata de múltiples variables, estas se muestreen con la misma frecuencia para facilitar su análisis y evitar la necesidad de técnicas de imputación, que podrían introducir distorsiones en los resultados.

1. Introducción

El objetivo: utilizar estos datos para aprender qué ocurrirá en instantes futuros, realizando predicciones en un horizonte concreto sobre el que es desconocido su comportamiento. Sin embargo, es importante destacar que no todos los fenómenos pueden ser predichos aunque dispongamos de un conjunto de datos suficiente y adecuado. Es esencial que, para que la técnica sea efectiva, estemos ante una tarea de forecasting, es decir, de predicción de elementos futuros, pero pudiéndonos apoyar en información pasada para comprender el fenómeno que estamos estudiando. Por ejemplo, tratar de predecir la temperatura que hará en las próximas 2 horas, en intervalos de 15 minutos, es una tarea viable: disponemos de gran cantidad de recursos históricos, como la tendencia de las horas anteriores, y los valores históricos de días previos, o incluso, el comportamiento en años anteriores, que pueden ser útiles.

Pero también podríamos enfrentarnos a escenarios más complejos, como el que gestiona diariamente Red Eléctrica para estimar el consumo de electricidad y responder con los tipos de energía más adecuados, evitando tanto el exceso como el déficit en la red. Para ello, se apoyan no solo en datos históricos de consumo, sino también en la información en tiempo real proveniente de sensores y mediciones realizadas en distintas estaciones eléctricas, lo que permite anticipar fluctuaciones y minimizar problemas de sincronización. A esto se suman otros factores externos que pueden influir, como las condiciones meteorológicas o incidencias en redes interconectadas de países vecinos. Tras la experiencia vivida en abril de 2025, queda en evidencia la importancia de esta tarea, ya que en caso de error, las consecuencias pueden ser muy graves: personas atrapadas en ascensores, hospitales en emergencia, problemas de tráfico, interrupción de la cadena de frío en los alimentos... Las pérdidas económicas se estimaron entre 1.600 y 2.500 millones de euros.

Por tanto, disponer de buenas técnicas de estudio para series temporales es una herramienta clave en multitud de aplicaciones. Manualmente, podemos realizar aproximaciones posibles para casos sencillos, donde apreciamos un comportamiento repetitivo en el tiempo de la serie (estacionalidad) o un comportamiento estrictamente creciente o decreciente linealmente (tendencia). Sin embargo, cuando el problema se vuelve más complejo o involucra múltiples variables, recurrir a matemáticas simples o técnicas intuitivas ya no es suficiente. En estos casos, es necesario contar con herramientas más sofisticadas que permitan modelar el comportamiento de la serie de forma aproximada, pero lo más fiel posible a la realidad.

Es aquí donde entran en juego métodos ampliamente utilizados, como la descomposición de la serie en componentes fundamentales: tendencia, estacionalidad y ruido. Una de las técnicas estadísticas más reconocidas en este ámbito es ARIMA (AutoRegressive Integrated Moving Average) [2]. Se basa en utilizar una ventana deslizante sobre la que ir calculando una media móvil, es decir, un intervalo de amplitud q entorno a un valor central, el cual se va moviendo de izquierda a derecha para evaluar con todas las instancias de la serie, y en una componente autoregresiva, que trata de incorporar información de instantes anteriores al que se está evaluando (hasta p valores). También se integra la serie tantas veces como sea necesario para suavizar su comportamiento. De este modo, es posible generar predicciones razonables, siempre y cuando el problema cumpla con ciertas condiciones que abordaremos más adelante.

No obstante, incluso técnicas como ARIMA presentan limitaciones importantes, especialmente cuando se enfrentan a horizontes de predicción amplios. A medida que intentamos anticipar valores más alejados en el tiempo, el modelo se ve obligado a basarse cada vez más

en sus propias predicciones previas en lugar de los datos observados, lo que provoca una acumulación progresiva de errores.

Esto plantea la necesidad de modelos más robustos, capaces de capturar dependencias de largo alcance y manejar múltiples variables de forma conjunta, tratando de minimizar el error evitando su acumulación. Podemos recurrir, como alternativa, a una de las herramientas ya empleadas en los grandes modelos de lenguaje que mencionamos anteriormente: los Transformers [6]. Pero su uso no es directo; debemos de adaptar su funcionamiento a las series temporales, ya que estos modelos, originalmente diseñados para tareas de lenguaje natural, no conservan de manera adecuada la información temporal del orden de los datos.

1.1.2. La dificultad de predicción a largo plazo: información posicional

Para poder adaptar modelos basados en Transformers para lenguaje natural a series temporales, es imprescindible introducir mecanismos de codificación posicional (positional encoding) que permitan al modelo interpretar correctamente la secuencia en el tiempo [4]. Sin este componente, los Transformers serían incapaces de distinguir el orden de la entrada, lo cual es esencial para predecir correctamente la evolución de un fenómeno.

Encontrar una codificación eficiente, sencilla y coherente con la estructura de los datos se encuentra en continua búsqueda en el estado del arte, pero la mayoría de ellas presentan los siguientes inconvenientes:

- **Falta de captura de la estructura.** En muchas ocasiones, se opta por utilizar codificaciones sencillas basadas en senos y cosenos, de la misma forma que se hace en procesamiento de lenguaje, pero de esta forma, estamos perdiendo información de patrones estacionales e información local de interés. Esto ocurre, por ejemplo, en modelos como Informer [7], uno de los primeros en adaptar los Transformers a series temporales.
- **Dificultad para adaptarse a diferentes escalas temporales y falta de semántica.** Propuestas simples, como la ya mencionada codificación sinusoidal, pueden ser ineficientes cuando se busca trabajar con distintas granularidades temporales. Al tratarse de una codificación fija, sus valores no se ajustan al cambiar la frecuencia de muestreo (por ejemplo, de segundos a minutos) ya que mantienen una amplitud y periodicidad predefinidas. Por otro lado, aproximaciones más complejas, como las basadas en convoluciones, intentan capturar patrones locales dentro de la secuencia, pero tienden a centrarse en un corto plazo, lo que dificulta la detección de relaciones de largo alcance o multiescala. Además, este tipo de codificación no incorpora explícitamente la semántica temporal (como la hora del día o el día de la semana), lo que limita su interpretabilidad y generalización en tareas donde esa información es relevante.
- **Complejidad.** Para paliar las dificultades de los métodos más simples, se han evaluado alternativas más complejas, que permiten aumentar el rendimiento: es el caso de Autoformer, que incorpora información autoregresiva al modelo, tomando así cierta similitud con la componente AR de ARIMA; o bien, se proponen encodings basados en transformadas de Fourier. Pero estas aumentan la necesidad de cómputo y el tiempo necesario para su entrenamiento, lo cual hace surgir otra vertiente alternativa, basada en modelos más sencillos, como Reformer [5], que trata de convertir la eficiencia cuadrática de los Transformers, a lineal. O Informer, que persigue lograr eficiencia

1. Introducción

$n \log n$ mediante su ProbSparse Attention [7]. Pero, a veces puede conseguirse el efecto contrario: empeorar en exceso los resultados, a consta de un menor tiempo de entrenamiento e inferencia. Por ejemplo, en Informer, al muestrearse en atención solo ciertos vectores de entrada, perdemos información local que puede ser clave.

La misión de este trabajo se centra en tratar de encontrar una alternativa que sea capaz de mantenerse cercana a la semántica del problema, permitiendo adaptabilidad en cuanto a parámetros, y tratando de alcanzar un equilibrio entre eficiencia y rendimiento.

1.2. Justificación

Tras estudiar el problema, surge la necesidad de crear una alternativa adecuada a los modelos de codificación existentes, incorporando dentro de ella la semántica del propio problema, e información local que permita una mayor adaptabilidad de los modelos creados, reduciendo las tasas de error. Aunque este objetivo ya es tratado por diversos modelos y líneas de investigación del estado del arte, en este trabajo buscamos replantear el positional encoding para que sea adaptable y capaz de incorporar información global y local, dando mayor importancia a una u otra en función de la entrada dada.

Todo esto se plantea aprovechando el conocimiento adquirido a partir de técnicas más clásicas, como ARIMA, e incorporando información estadística que resulte comprensible para prácticamente cualquier persona que utilice el modelo. De este modo, aunque se recurra posteriormente a modelos profundos como los Transformers, la codificación posicional introducida al inicio en los datos puede ser ajustable y fácilmente interpretable en función del contexto del problema.

Hasta ahora, gran parte de los esfuerzos se han centrado en adaptar soluciones desarrolladas originalmente para otros dominios, principalmente el procesamiento de lenguaje natural, al contexto de las series temporales. Sin embargo, estas adaptaciones no siempre se alinean completamente con las particularidades de los datos temporales, lo que puede derivar en pérdidas de información relevantes o en un rendimiento subóptimo. Esta desconexión pone de relieve la necesidad de desarrollar codificaciones específicamente diseñadas para capturar la estructura temporal propia de este tipo de datos. En este documento, abordaremos esta problemática desde una perspectiva crítica, comparando distintos métodos existentes y proponiendo nuevas soluciones, algunas de ellas híbridas, que buscan mejorar la representación temporal en modelos basados en Transformers.

1.3. Objetivos

Teniendo en cuenta los conceptos descritos anterioremente, y analizando el estado del arte en la actualidad, vemos justificada la necesidad la encontrar nuevas codificaciones posicionales con el objetivo de conseguir:

1. Realizar una revisión exhaustiva del estado del arte sobre positional encoding y su aplicación en modelos Transformer para series temporales, estudiando el funcionamiento de las diferentes técnicas y extrayendo el conocimiento útil para crear nuevas alternativas.

2. Analizar las propuestas originales de positional encoding adaptadas a series temporales, realizando un análisis crítico y explorando las diferentes vías alternativas.
3. Evaluar sistemáticamente la efectividad de los positional encoding estándar en tareas de forecasting sobre benchmarks de series temporales, haciendo uso de varios conjuntos de datos de diferente procedencia y estructura.
4. Establecer criterios claros para el diseño de nuevas codificaciones que permitan encontrar nuevas formas de incorporar información útil, pero además sirvan como guía para investigaciones futuras, identificando buenas prácticas, limitaciones comunes y condiciones bajo las cuales cada enfoque resulta más adecuado.
5. Comparar el impacto de las distintas codificaciones posicionales no solo en la precisión de las predicciones, sino también en aspectos como la capacidad de generalización y su eficiencia computacional.

En este documento, se recoge el estudio del estado del arte actual, realizando un estudio crítico y comparativo de las diferentes propuestas, haciendo también hincapié en el propio funcionamiento de los Transformer y como afecta a la codificación y procesado de la información. Se buscará, como ya se ha mencionado, crear un resumen con diferentes técnicas, nuevas y otras conocidas, que por separado o en su conjunto, de forma híbrida, propongan soluciones adecuadas a diferentes conjuntos de datos provenientes del mundo real.

Adicionalmente, se proporciona un repositorio en GitHub¹ que contiene todo el código desarrollado, así como las referencias a los conjuntos de datos utilizados. Este recurso se mantiene accesible con el objetivo de facilitar su análisis, reutilización y posible mejora en trabajos futuros.

¹<https://github.com/hexecoded/TFM>

2. Tendencias y Estado del arte

Antes de adentrarnos de lleno en el análisis del problema y la realización de nuevas metodologías, es necesario estudiar y comprender el estado actual de esta temática en el estado del arte. Debido a que no se ha visto tan explotada como otros ámbitos de la IA, como es el caso de los modelos convolucionales o el aprendizaje automático supervisado tabular, podemos encontrar continuos cambios y nuevas vertientes que pueden inspirarnos a la hora de abordar el problema. Además, de manera adicional, podremos ver si algunas de nuestras ideas ya han sido implementadas y dado resultado, ya que en caso negativo, podríamos evitar cometer los mismos errores y centrarnos en una vertiente mucho más prometedora.

Hasta el momento, buena parte de las técnicas diseñadas especialmente para series temporales y predicción a largo plazo, se basan principalmente en la descomposición las mismas en subcomponentes mucho más sencillas de procesar, las cuales pueden seguir un enfoque similar a divide y vencerás: descomponer la red en diferentes elementos, y mediante un mecanismo de agregación (aditivo, multiplicativo, o estadístico), recomponer la solución a la tarea. Sin embargo, se han producido solo avances menores en los últimos años en dicha dirección.

Actualmente, la principal tendencia en el modelado de series temporales se basa en la adaptación de modelos desarrollados para otras modalidades. Por ejemplo, se han incorporado convoluciones, propias de visión, con el objetivo de capturar patrones locales. Pero, lo más frecuente es encontrar adaptaciones propias de mecanismos de procesamiento de lenguaje natural, como los ya mencionados Transformers, debido a su capacidad para modelar dependencias de largo alcance de forma eficiente. Previamente, también se han explorado arquitecturas recurrentes, como las Recurrent Neural Networks (RNN) y sus variantes más avanzadas, como las Long Short-Term Memory (LSTM) [3], aunque estas presentaban ciertas limitaciones en cuanto a capacidad de paralelización y manejo de relaciones temporales de largo plazo.

A continuación, nos adentraremos de lleno en dichas propuestas, con el fin de aclarar, en primer, los conceptos clave acerca de las series temporales, y además, entender las problemáticas que surgen durante su procesamiento.

2.1. Procesamiento básico de Series Temporales

En la introducción, se han presentado brevemente las características fundamentales de las series temporales: su estructura temporal, caracterizada por muestreo, la importancia del orden de medición, y la necesidad de mantener las relaciones temporales para lograr aprender de forma efectiva.

Pero, en esta sección, profundizaremos en los aspectos clave del preprocesamiento necesario para su análisis de manera más formal, ya que a diferencia de otros tipos de datos, las

2. *Tendencias y Estado del arte*

series temporales requieren una atención especial a la dimensión temporal, y cualquier alteración en su estructura puede afectar directamente la capacidad del modelo para aprender los patrones subyacentes en ella. Por ello, es fundamental centrarnos en cuestiones como la frecuencia de muestreo, la consistencia temporal, el tratamiento de valores faltantes, y la normalización de las variables.

2.1.1. Temporalidad: estructura y frecuencia de muestreo

2.1.2. Valores faltantes

2.1.3. Normalización de los datos

2.1.4. Métodos para partición y evaluación del rendimiento

2.2. Modelos basados en descomposición

2.3. Transformers y Positional Encoding

2.4. Conjuntos de datos disponibles

3. Selección y preprocesado de los conjuntos de datos

4. Modelos de encoding posicional y entorno de trabajo

5. Análisis comparativo de positional encoding sobre las bases de datos

6. Conclusiones y trabajos futuros

A. Ejemplo de apéndice

Los apéndices son opcionales.

Este fichero `apendice-ejemplo.tex` es una plantilla para añadir apéndices al TFG. Para ello, es necesario:

- Crear una copia de este fichero `apendice-ejemplo.tex` en la carpeta `apendices` con un nombre apropiado (p.e. `apendice01.tex`).
- Añadir el comando `\input{apendices/apendice01}` en el fichero principal `tfm.tex` donde queremos que aparezca dicho apéndice (debe de ser después del comando `\appendix`).

Glosario

La inclusión de un glosario es opcional.

Archivo: `glosario.tex`

\mathbb{R} Conjunto de números reales.

\mathbb{C} Conjunto de números complejos.

\mathbb{Z} Conjunto de números enteros.

Bibliografía

- [1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang. On the opportunities and risks of foundation models, 2022.
- [2] G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc., USA, 1990.
- [3] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [4] H. Irani and V. Metsis. Positional encoding in transformer-based time series models: A survey, 2025.
- [5] N. Kitaev, Łukasz Kaiser, and A. Levskaya. Reformer: The efficient transformer, 2020.
- [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.
- [7] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting, 2021.