

---

# Clase 136 — Forecasting de series con RNN

Parte: 2 — Deep Learning · Fuente: Géron, cap. 15 § Forecasting a Time Series. Duración estimada: 80 min.

## Clase 136 — Forecasting de series con RNN

Parte: 2 — Deep Learning · Fuente: Géron, cap. 15 § Forecasting a Time Series. Duración estimada: 80 min.

### Objetivo

Aplicar RNN/LSTM/GRU a un problema real de forecasting de series temporales. Hacer split temporal (NO aleatorio), preparar windows, comparar contra baselines (naive, MA, ARIMA), y reportar métricas estándar (MAE, MAPE, RMSE).

### Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Hacer split temporal (train: período antiguo, val/test: período más reciente).
- Construir samples de longitud T con `tf.keras.utils.timeseries_dataset_from_array`.
- Comparar contra el baseline naïve ( $\hat{y}_t = y_{t-1}$ ) y media móvil.
- Reportar MAE, MAPE, RMSE.
- Reconocer cuándo Deep Learning es mejor que ARIMA / Prophet / XGBoost para series.

### Temas

- Split temporal estricto (no shuffle).
- Stationarity / diferenciación.
- Baseline naïve y por qué siempre comparar contra él.
- Windowing: cómo elegir tamaño de window (T) y horizonte.
- Forecasting multi-step: directo vs recursivo vs seq2seq.

### Definiciones y características

- Window / lookback: cuántos pasos pasados usás para predecir.
- Horizon: cuántos pasos hacia el futuro predecís.
- Naïve forecast: predicción = último valor observado.
- MAPE:  $\text{mean}(|y - \hat{y}| / |y|) \times 100$  — error porcentual.
- `timeseries_dataset_from_array`: helper Keras para crear ds de windows automáticamente.

### Dataset / recursos

- seaborn flights dataset o cualquier serie pública (e.g., consumo eléctrico).
- `tfds.load('electricity_load_diagrams')` o sintético.
- Librerías: tensorflow, keras, pandas, matplotlib.

### Ejercicios

1. Split temporal: separar primer 70 % train, 15 % val, último 15 % test. No mezclar.

2. Baseline naïve: `y_pred = y_test.shift(1)`. Reportar MAE.
3. LSTM forecasting: `Sequential([LSTM(32), Dense(1)])`. Comparar contra naïve.
4. GRU: igual con GRU. Comparar performance y velocidad.
5. Multi-step: predecir 7 pasos directamente (`Dense(7)` al final). Comparar contra predicción recursiva.

## Homework verificable

Forecasting de consumo eléctrico:

1. Dataset diario, 5 años; split 70/15/15 temporal.
2. Baselines: naïve y MA(7).
3. LSTM [64], lookback=30 días, horizon=7 días.
4. Reportar MAE en test para los 3.

Criterio de aceptación: LSTM debe superar a naïve en MAE; comparable o mejor que MA(7). Documentar dónde gana y dónde pierde.

## Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
<code>train_test_split</code> con shuffle por costumbre	Filtración de futuro al pasado. Fix: split
LSTM mejor que naïve por 0.001 → declarar	Sin significancia clara. Fix: usar pingou
Normalizar usando stats de todo el dataset	Leakage. Fix: fitear solo en train.
Predicción multi-step recursiva diverge	Error acumulado. Fix: entrenar con teacher
Reportar MAPE cuando hay valores cerca de	MAPE explota. Fix: usar SMAPE o reportar M

## Preguntas frecuentes

¿DL siempre supera a ARIMA?

No. Para series cortas, regulares, sin features exógenas: ARIMA / ETS suelen igualar o ganar. DL gana con series largas + features adicionales + complejidad no lineal.

¿Lookback óptimo?

Empezar con 1-2 ciclos (e.g., 14 días si hay estacionalidad semanal). Tunear con val.

¿RNN o Transformer para series?

Transformers de series temporales (TFT, Informer, PatchTST 2023) son state-of-the-art para series largas con muchas features. Para series chicas, LSTM/GRU + features manuales gana.

¿Multi-step directo o recursivo?

Directo: 1 modelo predice todos los horizontes. Recursivo: feedback de cada paso. Directo más simple y a veces mejor en horizonte largo.

¿Cómo manejo estacionalidad?

Features explícitas (mes, día de semana, hora) ayudan mucho. Diferenciar la serie también.

## Referencias

- Géron, cap. 15 — Forecasting a Time Series.
- Hyndman & Athanasopoulos (2021), Forecasting: Principles and Practice (libro gratuito).
- Lim et al. (2021), Temporal Fusion Transformers, IJF.
- Nie et al. (2023), PatchTST, ICLR.

## Siguiente clase

Clase 137 — LSTM, GRU

## Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

## Archivos complementarios

- notebook.ipynb