Día 06 · Tema 2 — Datos secuenciales: RNN, LSTM y GRU

Eduardo C. Garrido-Merchán · ecgarrido@comillas.edu Universidad Pontificia Comillas · ICADE

1. Datos secuenciales

Una serie temporal es una secuencia $\{x_t\}_{t=1}^{T}$ con $x_t \in \mathbb{R}^d$, donde el orden importa: x_t depende del pasado $x_{<t}$ y no es i.i.d. Forecast significa modelar la distribución condicional

$$p\left(x_{t+1} \,\middle|\, x_{1:t}\right). \tag{1}$$

En retail (Inditex, Mercadona), x_t son ventas semanales por SKU; en energía (Iberdrola), MWh por hora.

2. La RNN básica

La red neuronal recurrente comprime todo el pasado en un *estado oculto* $h_t \in \mathbb{R}^{d_h}$:

$$h_t \;=\; \sigma\big(W_h\, h_{t-1} \;+\; W_x\, x_t \;+\; b\big), \qquad \hat{y}_t \;=\; W_o\, h_t + b_o. \tag{2}$$

$h_t \in \mathbb{R}^{d_h}$	estado oculto, memoria *viajera*
$x_t \in \mathbb{R}^{d}$	entrada en el paso t
$W_h \in \mathbb{R}^{d_h \times d_h}$	matriz recurrente
$W_x \in \mathbb{R}^{d_h \times d}$	matriz de entrada
σ	activación, típicamente tanh
W_o, b_o	capa de salida

Tres pasos de RNN sobre demanda $x_1 = 10$, $x_2 = 12$, $x_3 = 11$ con $h_0 = 0$ y matrices escalares $w_h = 0{,}6$, $w_x = 0{,}4$, $b = 0$, tanh: $h_1 = \tanh(0{,}4 \cdot 10) = 0{,}999$, $h_2 = \tanh(0{,}6 \cdot 0{,}999 + 0{,}4 \cdot 12) = 0{,}999, \ldots$ La RNN comprime el pasado en un único número (en este ejemplo a escalar).

3. Backpropagation through time

El gradiente de la pérdida en el paso t respecto al estado h_k ($k < t$) es

$$\frac{\partial \ell_t}{\partial h_k} \;=\; \frac{\partial \ell_t}{\partial h_t} \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \;=\; \frac{\partial \ell_t}{\partial h_t} \prod_{j=k+1}^{t} W_h^{\top}\, \mathrm{diag}(\sigma'(\cdot)). \tag{3}$$

El producto $\prod W_h^{\top}$ explota si $\|W_h\|_2 > 1$ y se desvanece si $\|W_h\|_2 < 1$. Es la causa estructural de que las RNN vanilla no aprendan dependencias largas.

4. LSTM: cuatro puertas controlan la memoria

La LSTM separa *célula* C_t (memoria) y estado oculto h_t, con cuatro puertas:

$$\begin{aligned}
f_t &= \sigma\big(W_f [h_{t-1}, x_t] + b_f\big) && \text{(forget)} \\
i_t &= \sigma\big(W_i [h_{t-1}, x_t] + b_i\big) && \text{(input)} \\
\tilde{C}_t &= \tanh\big(W_C [h_{t-1}, x_t] + b_C\big) && \text{(candidate)} \\
o_t &= \sigma\big(W_o [h_{t-1}, x_t] + b_o\big) && \text{(output)} \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad h_t = o_t \odot \tanh(C_t). &&
\end{aligned} \tag{4}$$

$f_t, i_t, o_t \in [0,1]^{d_h}$	puertas; σ es sigmoide
$\tilde{C}_t$	candidato de actualización para la célula
C_t	célula: la memoria de largo plazo
$[h_{t-1}, x_t]$	concatenación

La célula C_t se propaga con **operaciones lineales**: el gradiente fluye sin atenuación a través de $\prod f_j$, no a través de Jacobianos. Por eso la LSTM captura dependencias largas.

5. GRU: dos puertas, ligeramente menos

La GRU fusiona forget e input en una sola puerta y elimina la célula:

$$\begin{aligned} z_t &= \sigma(W_z[h_{t-1}, x_t]) && \text{(update)} \\ r_t &= \sigma(W_r[h_{t-1}, x_t]) && \text{(reset)} \\ \tilde{h}_t &= \tanh\big(W_h[r_t \odot h_{t-1},\ x_t]\big) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \end{aligned} \tag{5}$$

Menos parámetros (3 matrices W en lugar de 4), **calidad similar** en la mayoría de tareas; preferida cuando el dataset es pequeño.

6. Forecast multipaso y métricas

Si se desea pronosticar h pasos $\hat{x}_{t+1:t+h}$, hay dos estrategias: **recursiva** (alimentar $\hat{x}_{t+1}$ como entrada para predecir $\hat{x}_{t+2}$, etc.) y **directa** (entrenar una salida por horizonte $\hat{x}_{t+k}$ para cada $k = 1, \ldots, h$). La métrica estándar de retail es MAPE:

$$\text{MAPE} \;=\; \frac{100\,\%}{Nh} \sum_{t=1}^{N} \sum_{k=1}^{h} \left| \frac{x_{t+k} - \hat{x}_{t+k}}{x_{t+k}} \right|. \tag{6}$$

Idea clave. En muchos casos un LSTM no supera a ARIMA/ETS hasta no añadir features exógenas: precio, promoción, calendario, clima. Si no las tienes, ARIMA es la opción honesta.