Día 05 · Tema 2 — Transfer learning en profundidad y el Hub de Hugging Face

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

1. Por qué transfer gana con poco dato

Sea f_{θ_0} preentrenado en un dataset enorme $\mathcal{D}_{\text{src}}$ (p.ej. ImageNet, $1{,}3 \cdot 10^6$ imágenes). El error en el dataset objetivo $\mathcal{D}_{\text{tgt}}$ se descompone como

$$\mathbb{E}\big[\hat{\mathcal{R}}(\hat{\theta})\big] \;\approx\; \underbrace{\hat{\mathcal{R}}_{\text{tgt}}(\theta^\star)}_{\text{error irreducible}} \;+\; \underbrace{\mathcal{O}\!\left(\sqrt{\tfrac{\mathrm{VC}(\mathcal{F})}{n}}\right)}_{\text{generalización}} \;+\; \underbrace{d(\mathcal{D}_{\text{src}}, \mathcal{D}_{\text{tgt}})}_{\text{transfer gap}}. \tag{1}$$

- $\hat{\mathcal{R}}$ riesgo empírico en el dataset objetivo
- $\mathrm{VC}(\mathcal{F})$ capacidad efectiva del espacio de modelos $\mathcal{F}$
- n tamaño de $\mathcal{D}_{\text{tgt}}$ etiquetado
- $d(\cdot,\cdot)$ discrepancia entre dominios (p.ej. $\mathcal{H}\Delta\mathcal{H}$-distance)

Con poco n el término de generalización domina; preentrenar baja la capacidad efectiva al restringir $\mathcal{F}$ alrededor de θ_0.

2. Los tres niveles de transfer, formalmente

Sea $\theta = (\theta_b, \theta_h)$ con θ_b el backbone y θ_h la cabeza. Los tres regímenes son:

$$\begin{aligned}
&\text{(L1) Feature extraction:} && \hat{\theta}_h = \arg\min_{\theta_h} \hat{\mathcal{R}}(\theta_b^{(0)}, \theta_h), \\
&\text{(L2) Fine-tune ligero:} && \hat{\theta} = \arg\min_{(\theta_b,\theta_h)} \hat{\mathcal{R}}(\theta), \quad \eta_b \ll \eta_h, \\
&\text{(L3) Fine-tune completo:} && \hat{\theta} = \arg\min_{(\theta_b,\theta_h)} \hat{\mathcal{R}}(\theta), \quad \eta_b = \eta_h.
\end{aligned} \tag{2}$$

Regla práctica: $n < 10^3 \Rightarrow$ L1; $10^3 < n < 10^4 \Rightarrow$ L2; $n \geq 10^4 \Rightarrow$ L3.

3. Discriminative learning rates

En L2 conviene tasas de aprendizaje por bloque, decreciendo hacia las capas tempranas:

$$\eta_\ell \;=\; \eta_0 \cdot r^{L-\ell}, \qquad r \in (0,1). \tag{3}$$

- η_ℓ learning rate de la capa ℓ
- η_0 LR de la cabeza
- r factor de decaimiento (p.ej. 0,5 ó 0,1)
- L índice de la última capa

Intuición: las capas tempranas aprenden bordes y texturas (universales); las tardías aprenden objetos específicos del dominio fuente y deben adaptarse más.

4. La receta exprés (seis pasos)

1. Cargar modelo preentrenado: `AutoModelFor*.from_pretrained("...")`.
2. Sustituir cabeza por una con `num_labels` del dominio objetivo.
3. Congelar backbone, entrenar 1–3 épocas para inicializar bien la cabeza.
4. Descongelar, aplicar *discriminative LR*.
5. Entrenar 5–20 épocas con `Trainer` + early stopping.
6. Evaluar en test *retenido* y guardar el modelo en `push_to_hub`.

Idea clave. Las seis líneas críticas en código no exceden ~15: cargar modelo, cargar dataset, definir `TrainingArguments`, definir métricas, crear `Trainer`, llamar a `trainer.train()`.

5. Métricas de selección en el Hub

El Hub de Hugging Face permite filtrar por (i) licencia, (ii) tamaño en GB, (iii) número de descargas y (iv) métrica en benchmark. Para un caso de negocio en castellano sobre OCR de facturas:

- `microsoft/dit-base-finetuned-rvlcdip`: 86 M parámetros, ~340 MB.
- `naver-clova-ix/donut-base`: 200 M, ~700 MB.
- `microsoft/layoutlmv3-base`: 133 M, ~500 MB.

La decisión se toma por la combinación (latencia objetivo → GB máximos) × (calidad mínima → benchmark en el Hub).

6. Coste-beneficio del transfer (ejemplo BBVA OCR)

Con $n = 1500$ facturas etiquetadas y un θ_0 de DiT preentrenado, el F1 alcanzable en el dominio destino sigue la ley empírica

$$F_1(n) \;\approx\; F_1(\infty)\left(1 - A\,n^{-\alpha}\right), \qquad \alpha \in [0{,}3,\, 0{,}7]. \tag{4}$$

$F_1(\infty)$ plateau alcanzable con datos infinitos
A constante específica del dominio
α exponente de la curva de aprendizaje

Empíricamente: pasar de $n = 300$ a $n = 1500$ sube F1 de 0,78 a 0,90; pasar a $n = 15000$ apenas a 0,93. Decisión: invertir en datos hasta el codo de la curva, no más allá.