Clase 074 — Early stopping

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 4. Duración estimada: 45 min.

Clase 074 — Early stopping

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 4. Duración estimada: 45 min.

Objetivo

Aplicar early stopping como técnica de regularización implícita en entrenamientos iterativos: monitorear la pérdida de validación durante el descenso por gradiente y detener el ajuste cuando deja de mejorar, conservando el mejor modelo visto.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Explicar por qué detener el entrenamiento antes del óptimo de train actúa como regularización.
- Configurar SGDRegressor(early_stopping=True) con validation_fraction, n_iter_no_change y tol.
- Graficar curvas de train loss vs. validation loss e identificar la best epoch.
- Implementar manualmente un loop con paciencia y snapshot del mejor modelo (partial_fit).
- Decidir cuándo early stopping reemplaza o complementa a Ridge/Lasso.

Temas

- Sobreajuste en entrenamientos iterativos (SGD, gradient boosting, redes neuronales).
- Curva de aprendizaje por época: train baja, validación baja y luego sube.
- Mecánica del early stopping: monitorear val loss + criterio de paciencia.
- API de scikit-learn: SGDRegressor / SGDClassifier con early_stopping=True.
- Implementación manual con partial_fit + copy.deepcopy del mejor estimador.
- Relación con otras regularizaciones (L1, L2, dropout en deep learning).

Definiciones y características

- Early stopping: detener el entrenamiento iterativo cuando una métrica de validación deja de mejorar durante un número fijo de iteraciones, devolviendo los parámetros del mejor punto observado.
- Validation loss: error medido sobre un split separado del de entrenamiento; es la señal que decide cuándo cortar. No debe usarse el test set para esta decisión.
- Patience (paciencia) / n_iter_no_change: cantidad de épocas consecutivas sin mejora superior a tol que se toleran antes de detener. Valores típicos: 5–20.
- Best epoch snapshot: copia de los parámetros (coef_, intercept_) en la época con menor val loss. Sin este snapshot, al cortar nos quedaríamos con un modelo ya degradado.
- tol: umbral mínimo de mejora para considerar que hubo progreso. Si loss_actual > loss_best - tol, la época no cuenta como mejora.
- Regularización implícita: a diferencia de Ridge/Lasso (que penalizan la magnitud de los pesos en la función objetivo), early stopping limita cuánto se ajustan los pesos, controlando capacidad efectiva.

Dataset / recursos

- sklearn.datasets.fetch_california_housing para regresión.
- sklearn.datasets.make_regression(n_samples=2000, n_features=50, noise=20) para experimentos controlados.
- Géron, Hands-On ML (3ª ed.), cap. 4 § "Early Stopping" (figura de la "U" en val loss).

Ejercicios

1. Curva clásica: entrenar SGDRegressor(max_iter=1, warm_start=True, learning_rate='constant', eta0=0.0005) por 500 épocas sobre California Housing escalado. Graficar RMSE de train y validación por época. Marcar la best epoch.
2. Early stopping automático: comparar SGDRegressor(early_stopping=True, validation_fraction=0.2, n_iter_no_change=10, tol=1e-4) contra el modelo sin early stopping. Reportar n.º de épocas reales (n_iter_) y RMSE en test.
3. Paciencia: barrer n_iter_no_change ∈ {1, 5, 20, 100} y mostrar cómo afecta la época final y el error de test.
4. Implementación manual: escribir un loop con partial_fit que mantenga best_loss, best_model = deepcopy(sgd) y un contador de paciencia. Devolver el mejor modelo.
5. Comparación con Ridge: sobre make_regression con ruido, comparar (a) SGD sin regularización + early stopping, (b) Ridge con alpha tuneado por CV. Discutir cuál generaliza mejor y por qué.

Homework verificable

Sobre California Housing (split 60/20/20 train/val/test, features escaladas con StandardScaler):

1. Entrenar un SGDRegressor con early_stopping=True, validation_fraction=0.2, n_iter_no_change=15, tol=1e-4, random_state=42.
2. Guardar la curva de loss_curve reconstruida con partial_fit (paralela, sin early stopping, 1000 épocas) en curva_val.png.
3. Reportar: épocas usadas por el modelo con early stopping (n_iter_), RMSE en test, y época óptima vista en la curva manual.

Criterio de aceptación: RMSE_test ≤ 0.75 (en variable target sin escalar, en cientos de miles de USD), y la época con early stopping debe estar dentro de ±10 % de la best epoch detectada manualmente.

Errores comunes

- Monitorear train loss en vez de val loss: train siempre baja; nunca dispara el corte. Hay que usar un split de validación interno.
- No guardar el snapshot del mejor modelo: si cortás en la época k después de patience épocas malas, los pesos finales no son los mejores. SGDRegressor lo hace por dentro; en implementación manual hay que hacerlo explícito.
- Confundir validación con test: usar el test set para decidir el corte filtra información y arruina la estimación de generalización. El test se toca una sola vez al final.
- Patience demasiado baja: con ruido en val loss, n_iter_no_change=1 corta prematuro por fluctuaciones aleatorias. Subir a 5–20.
- Olvidar escalar features: SGD es muy sensible a la escala; sin StandardScaler el descenso oscila y la curva de validación no muestra la "U" clara.

Preguntas frecuentes

- ¿Early stopping reemplaza a Ridge/Lasso? No siempre. En modelos lineales con muchos features correlacionados, Ridge suele dar mejor sesgo-varianza. Early stopping brilla en modelos iterativos costosos (boosting, redes) donde tunear alpha por CV es caro.
- ¿Sirve para modelos cerrados como LinearRegression o Ridge con solver='cholesky'? No: esos resuelven en un paso, no hay "épocas" que detener. Aplica a algoritmos iterativos: SGD, gradient boosting (n_estimators con early_stopping_rounds), redes neuronales.
- ¿Cuánto debería ser validation_fraction? 10–20 % suele alcanzar. Con datasets chicos (<1000 filas) usar CV externa en lugar de un split interno fijo.
- ¿n_iter_no_change=5 es estándar? Es el default de sklearn y razonable. Subilo si la val loss es ruidosa o si la curva baja muy lento.
- ¿Por qué la val loss puede subir después de cierto punto? Porque el modelo empieza a memorizar ruido del train: train sigue bajando pero la capacidad efectiva se gastó en patrones espurios que no generalizan.

Referencias

- Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (3ª ed.), cap. 4, sección "Early Stopping".
- scikit-learn docs: SGDRegressor (parámetros early_stopping, validation_fraction, n_iter_no_change, tol).
- Prechelt, L. (1998). Early Stopping — But When? en Neural Networks: Tricks of the Trade.

Siguiente clase

Clase 075 — Regresión logística binaria y softmax

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb