
Clase 129 — Pooling

Parte: 2 — Deep Learning · Fuente: Géron, cap. 14 § Pooling Layers. Duración estimada: 45 min.

Clase 129 — Pooling

Parte: 2 — Deep Learning · Fuente: Géron, cap. 14 § Pooling Layers. Duración estimada: 45 min.

Objetivo

Conocer pooling — operación sin parámetros que reduce dimensiones espaciales: MaxPooling2D, AveragePooling2D, GlobalAveragePooling2D. Saber que max-pool agrega invariancia local a translación y reduce cómputo de capas posteriores. Comparar contra el approach moderno (stride > 1 en Conv).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Aplicar MaxPooling2D(2), AveragePooling2D(2), GlobalAveragePooling2D() y conocer sus shapes de salida.
- Diferenciar max-pool (preserva la característica más fuerte) de average-pool (promedio espacial).
- Aplicar GlobalAveragePooling2D antes de la cabeza Dense (estándar en CNN modernas — reemplaza Flatten).
- Reconocer que pooling no tiene parámetros entrenables (es solo una reducción).
- Saber que ResNet/ConvNeXt usan stride en lugar de pool intermedio.

Temas

- MaxPool: ventana $k \times k$ → toma el máximo. Default pool_size=2, strides=2 → halve H y W.
- AvgPool: promedio de la ventana. Suaviza.
- GlobalAvgPool: una sola activación por feature map → (batch, channels).
- Invariancia a translación local: si el feature se mueve 1 px, el max no cambia.
- Pool vs stride: equivalentes en muchos casos; tendencia moderna es eliminar pool intermedio.

Definiciones y características

- MaxPooling2D(pool_size=2, strides=2, padding='valid'): ventana 2×2 → 1 valor (el máximo).
- AveragePooling2D: como max pero con promedio.
- GlobalAveragePooling2D: agrega sobre toda la spatial dim → (batch, channels). Sin pool size.
- GlobalMaxPooling2D: similar, con max.
- Output shape: para pool_size=2 stride=2: (H/2, W/2, C).

Dataset / recursos

- Fashion-MNIST o CIFAR-10.
- Librerías: tensorflow, keras.

Ejercicios

1. MaxPool shape: aplicar MaxPool(2) a tensor (1, 28, 28, 32). Verificar salida (1, 14, 14, 32).

2. MaxPool vs AvgPool: con la misma CNN, intercambiar y comparar accuracy en Fashion-MNIST. MaxPool suele ganar marginal.
3. GlobalAvgPool reemplaza Flatten: arquitectura Conv → Conv → GlobalAvgPool → Dense(10) vs Conv → Conv → Flatten → Dense(128) → Dense(10). Comparar params y accuracy.
4. Pool vs stride: comparar Conv(stride=1) → Pool(2) vs Conv(stride=2) (sin pool). En modelos chicos son prácticamente equivalentes.
5. padding='same' en pool: comportamiento si H es impar. valid recorta, same agrega zeros.

Homework verificable

Comparar 3 arquitecturas sobre Fashion-MNIST:

1. Conv(32) → MaxPool → Conv(64) → MaxPool → Flatten → Dense(128) → Dense(10).
2. Conv(32) → MaxPool → Conv(64) → MaxPool → GlobalAvgPool → Dense(10).
3. Conv(32, stride=2) → Conv(64, stride=2) → GlobalAvgPool → Dense(10) (sin pool).

Reportar accuracy y # parámetros.

Criterio de aceptación: la opción 2 tiene muchos menos parámetros y accuracy similar; la 3 también es competitiva (validando el approach moderno).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
pool_size=3, strides=1 recorta poco	Posible pero raro. Fix: usual es pool=2, s
MaxPool antes del primer Conv	Pierde detalle al inicio. Fix: pool después
Olvido pool y modelo Dense tiene millones	Flatten de feature maps grandes. Fix: pool
padding='same' en pool con H impar genera	TF agrega un row/col de zeros. Fix: pre-cr
Usar pooling en NLP / Transformers	No tiene sentido. Fix: pool es espacial; e

Preguntas frecuentes

¿MaxPool o AvgPool?

MaxPool por default — captura "el más fuerte" (útil en clasificación). AvgPool en regresión espacial (segmentación final) o cuando el promedio es semánticamente relevante.

¿GlobalAvgPool obligatorio en ResNet?

Sí, es la última capa antes del FC final. Reduce el riesgo de overfit del Flatten + Dense gigante.

¿Pool intermedio aún se usa?

CNNs clásicas (VGG, LeNet) sí. ResNet/ConvNeXt usan stride en conv (sin pool intermedio).

¿pool_size=4 o más grande?

Raro. Casi siempre 2. Más grande pierde información rápido.

¿Pool diferenciable?

MaxPool tiene gradiente subdiferencial (= 1 en el max, 0 en el resto). Avg es diferenciable normal.

TF/PyTorch los manejan transparente.

Referencias

- Géron, cap. 14 — Pooling Layers.
- Keras MaxPooling2D.
- Springenberg et al. (2015), Striving for Simplicity: The All Convolutional Net — argumenta eliminar pool intermedio.

Siguiente clase

Clase 130 — Arquitecturas CNN: LeNet, AlexNet, VGG, GoogLeNet, ResNet, Xception, SENet, EfficientNet, ConvNeXt

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb