Agregación eficiente de características por caminos en Redes Convolucionales Residuales

Iosu Rodriguez-Martinez[1 [0000−0002−9960−0203]] and Humberto Bustince[1 [0000−0002−1279−6195]]

Universidad Pública de Navarra, Departamento de Estadística, Informática y Matemáticas, Pamplona C/Cataluña s/n, 31006 iosu.rodriguez@unavarra.es

Abstract. La fusión de características es un requisito habitual de las redes neuronales modernas, que a menudo requieren reducir características redundantes o combinar múltiples representaciones internas en una sola. En la mayoría de arquitecturas este problema se aborda mediante transformaciones afines entrenables, lo que incrementa su ya elevado número de parámetros. Nuestra hipótesis es que funciones de agregación con pocos parámetros pueden reemplazar estas operaciones de fusión, dando lugar a modelos más ligeros pero igualmente eficaces. En particular, en la arquitectura ResNeXt sustituimos la etapa de fusión de caminos de sus bloques convolucionales por una función de agregación seguida de un operador de convolución ligero como proyección final. Nuestros resultados preliminares indican que esta construcción resulta competitiva en conjuntos de datos de tamaño reducido pese a reducir notablemente el número de parámetros del modelo.

Keywords: Redes Neuronales Convolucionales · Redes multi-rama · Funciones de agregación.

1 Motivación

Las Redes Neuronales Convolucionales (CNNs) siguen siendo una familia dominante en visión artificial, especialmente atractiva en escenarios con recursos limitados gracias a sus sesgos inductivos y su eficiencia. No obstante, el elevado número de parámetros y el consumo de memoria y cómputo de los modelos actuales dificultan su despliegue en *hardware* limitado y restringen su escalabilidad.

Dentro de estas arquitecturas son frecuentes las operaciones de fusión de información. Clásicamente se realizaban mediante funciones sencillas y sin parámetros, como operadores de *pooling* para la reducción espacial o *global average pooling* antes del clasificador final [5]. Con el tiempo, muchas se han sustituido por alternativas parametrizadas (p. ej. convoluciones con *stride* $= 2$ en lugar de capas de *pooling*, o transformaciones afines aprendibles para reducir la dimensionalidad de los mapas de característica. Estos operadores mejoran la precisión a costa de introducir parámetros adicionales en los puntos de fusión de la red.

Un ejemplo representativo es la arquitectura ResNeXt [6], que reformula el bloque residual de ResNet [3] sustituyendo su único camino por k caminos paralelos de menor complejidad, cada uno con un número reducido de canales C_h

(k es la "cardinalidad" del bloque). Los canales de salida de todos los caminos se concatenan y se procesan mediante una última convolución 1×1 hasta los C_o canales deseados, fusionándolos y transformándolos de forma implícita y simultánea (Figura 1.a).

En este trabajo proponemos invertir el orden de las operaciones: agregar primero los k caminos elemento a elemento y aplicar después una única convolución 1×1, reduciendo los parámetros de esa proyección en un factor de k.

En esta propuesta inicial hemos considerado funciones entrenables pero ligeras procedentes de la teoría de la agregación, como las medias ponderadas y los operadores de media ponderada ordenada (*Ordered Weighted Averaging*, OWA) [7]. Nuestra hipótesis es que esta fusión basada en agregación puede recuperar gran parte de la capacidad representacional de la proyección parametrizada con una fracción de su coste. Nuestros resultados preliminares parecen apoyar esta hipótesis al trabajar con conjuntos de datos reducidos.

2 Metodología

2.1 Fusión de caminos basada en agregación

Una unidad *bottleneck* de ResNeXt está compuesta por tres convoluciones: una 1×1 que reduce la entrada a una anchura de $k\cdot C_h$ canales, una 3×3 que procesa los k caminos paralelos de anchura C_h cada uno, y una última 1×1 que fusiona los caminos y los proyecta a C_o canales de salida. Cada convolución va seguida de una capa de *batch normalization* y ReLU, con una conexión residual antes de la activación final. Nuestra modificación afecta a la última convolución 1×1, que combina los k caminos.

Sean $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^{C_h \times H \times W}$ las salidas de los caminos. El bloque estándar las combina mediante una única convolución 1×1 $W \in \mathbb{R}^{C_o \times k \cdot C_h}$ sobre los canales concatenados, lo que equivale a una suma de k proyecciones independientes por camino:

$$\mathbf{y} \;=\; W\,[\mathbf{x}_1 \oplus \cdots \oplus \mathbf{x}_k] \;=\; \sum_{j=1}^{k} W_j\,\mathbf{x}_j, \tag{1}$$

donde $\oplus$ es la concatenación de canales y $W_j \in \mathbb{R}^{C_o \times C_h}$ es el j-ésimo bloque de W. Esta proyección requiere $k \cdot C_h \cdot C_o$ parámetros. En su lugar, proponemos agregar los caminos elemento a elemento mediante funciones de agregación (a, b) [1] $A : [a, b]^k \to [a, b]$, con $a, b \in \mathbb{R}$, a lo largo de la dimensión de cardinalidad, antes de aplicar una única proyección compartida $W' \in \mathbb{R}^{C_o \times C_h}$ (Figura 1.b):

$$\mathbf{y} \;=\; W'\,\mathbf{A}(\mathbf{x}_1, \ldots, \mathbf{x}_k), \tag{2}$$

donde $\mathbf{A}$ representa la aplicación elemento a elemento de A por canal y por posición espacial entre los distintos caminos. La proyección requiere ahora $C_h \cdot C_o$ parámetros, una reducción de factor k en esta proyección.

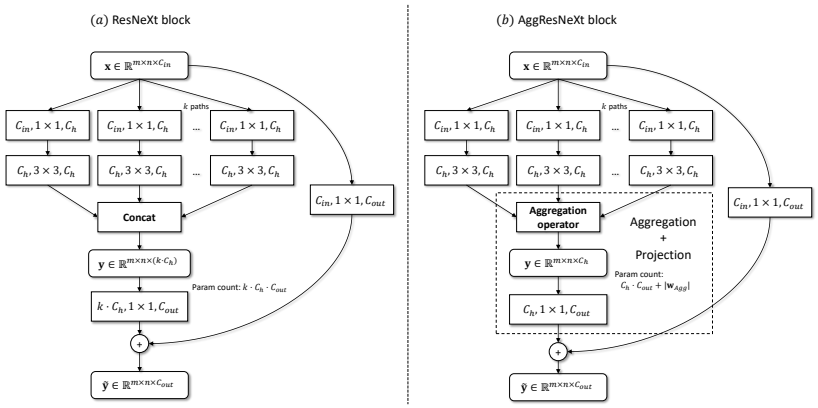


Fig. 1. Bloque ResNeXt (a) frente a nuestro bloque AggResNeXt (b). Sustituimos la convolución 1×1 de fusión por una agregación elemento a elemento sobre los caminos seguida de una proyección de dimensionalidad reducida, lo que reduce los parámetros de fusión en un factor de k.

En este trabajo preliminar hemos estudiado tres funciones de agregación (a, b) para A: la media aritmética $AM(\mathbf{x}) = \frac{1}{k}\sum_i x_i$; una media ponderada $WM(\mathbf{x}) = \sum_i w_i x_i$ $(w_j \geq 0, \sum_j w_j = 1)$; y un OWA $OWA(\mathbf{x}) = \sum_i w_i x_{\sigma(i)}$, donde $\mathbf{x}_\sigma$ denota la permutación decreciente de $\mathbf{x}$.

2.2 Configuración experimental

Evaluamos nuestra propuesta sobre CIFAR-10, CIFAR-100 [4] y Caltech-101 [2], con particiones 80/20. Las imágenes se redimensionan a 32×32 y, en entrenamiento, aplicamos como *data augmentation* un rellenado con ceros de 4 píxeles, un recorte aleatorio de 32×32 y un volteo horizontal aleatorio.

Todos los modelos comparten un *backbone* reducido de tipo ResNet/ResNeXt para entradas de 32×32: una convolución 3×3 inicial, tres etapas de 3, 4 y 6 unidades *bottleneck* con expansión 4, y una capa de *GAP* con clasificador lineal. Comparamos **ResNet** (unidades *bottleneck* de un solo camino), **ResNeXt** [6] (cardinalidad 32 y anchuras 4, 8 y 16 en sus tres etapas) y **AggResNeXt**, nuestra propuesta con distintas funciones de agregación (a, b).

Entrenamos 5 ejecuciones independientes por combinación de modelo y conjunto de datos, y reportamos la precisión en test como media $\pm$ desviación típica. Optimizamos mediante Descenso de Gradiente Estocástico durante 90 épocas, con la tasa de aprendizaje reducida por *Cosine Annealing* desde 0.0001 hasta 0.

3 Resultados parciales

La Tabla 1 recoge la tasa de acierto media en test. Pese a usar un 43% menos de parámetros, las variantes basadas en agregación se mantienen competitivas

Table 1. Tasa de acierto media en test (± desviación típica) en los distintos conjuntos de datos.

Modelo	CIFAR10	CIFAR100	CALTECH101
ResNet	0.9247 ± 0.0071	0.7323 ± 0.0057	0.5877 ± 0.0304
ResNeXt	0.9378 ± 0.0024	0.7379 ± 0.0126	0.6455 ± 0.0190
AM	0.9246 ± 0.0035	0.6958 ± 0.0067	0.6217 ± 0.0297
WM	0.9233 ± 0.0017	0.6905 ± 0.0053	0.6535 ± 0.0194
OWA	0.9202 ± 0.0025	0.6924 ± 0.0050	0.6590 ± 0.0186

frente a ResNeXt en el conjunto más sencillo, CIFAR-10, y superan claramente a ResNet en Caltech-101, donde los operadores entrenables (media ponderada y OWA) igualan a ResNeXt. En CIFAR-100, el más difícil, nuestra propuesta resulta menos favorable, unos pocos puntos por debajo de ambas líneas base.

De ello extraemos dos conclusiones. Primero, el coste de la reducción de parámetros crece con la dificultad de la tarea. Segundo, en el régimen de pocos datos de Caltech-101 la agregación resulta ventajosa, lo que sugiere un beneficio frente al *overfitting*. Estas observaciones respaldan nuestra hipótesis y motivan un estudio futuro sobre otras cardinalidades y funciones de agregación.

Acknowledgments. Este estudio ha sido financiado por el proyecto PID2022-136627NB-I00 financiado por MCIN/AEI/10.13039/501100011033/FEDER, UE.

Disclosure of Interests. Los autores declaran no tener intereses en competencia que sean relevantes para el contenido de este artículo.

References

1. da Cruz Asmus, T., Dimuro, G.P., Bedregal, B., Sanz, J.A., Fernandez, J., Rodriguez-Martinez, I., Mesiar, R., Bustince, H.: A constructive framework to define fusion functions with floating domains in arbitrary closed real intervals. Information Sciences **610**, 800–829 (2022)
2. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples... In: CVPR 2004 (2004)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
4. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
5. Lin, M., Chen, Q., Yan, S.: Network in network. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014), https://arxiv.org/abs/1312.4400
6. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
7. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on systems, Man, and Cybernetics **18**(1), 183–190 (1988)