Generalizing Average- and Max-Pooling

University of Basel

Florentin Bieder, Robin Sandkühler, Philippe C. Cattin

Center for medical Image Analysis & Navigation

Department of Biomedical Engineering

What are Pooling Blocks?

In convolutional neural networks (CNNs) we frequently need to perform a reduction of the spatial resolution of the feature maps. Two of the most popular methods are *max-pooling* and *average-pooling*.

Pooling methods are usually implemented as *sliding window* methods: A "window" is being slid across the input feature map, and for each location, the values within that window are passed to some function f which computes the corresponding value for the output feature map.

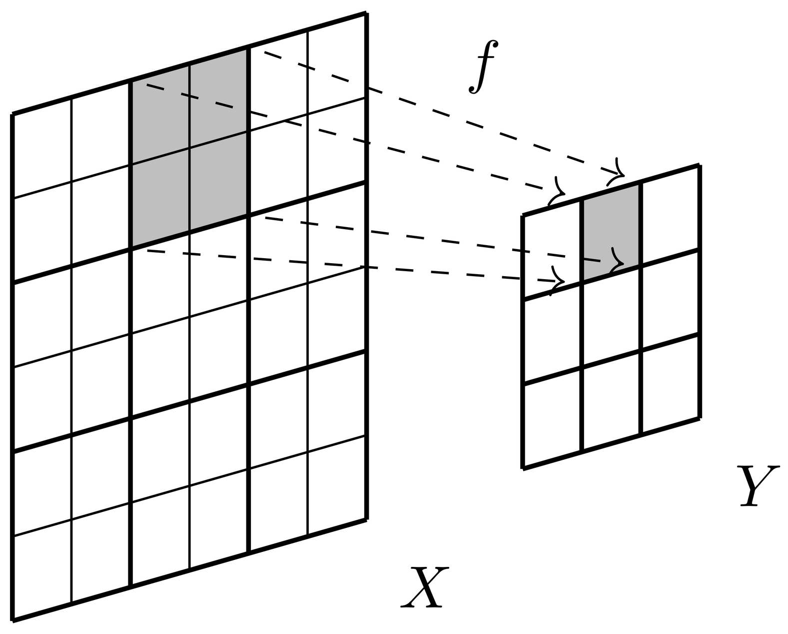

Fig. 1: Visualization of pooling with stride 2 and a window size of 2×2. This means $f : \mathbb{R}^n \to \mathbb{R}$ with $n = 4$.

A frequent choice of f is the maximum or alternatively the average, resulting in *max-pooling* or *average-pooling*.

$$MP(\vec{x}) = \max_i x_i \qquad AP(\vec{x}) = \frac{1}{n}\sum_i x_i$$

But there are many, more interesting choices of functions that generalize both of these.

Generalizing Average- and Max-Pooling

Some examples that generalize both average- and max-pooling include:

Gated Pooling was introduced by Lee et al. [1] and is defined by

$$GP(\vec{x}) = g\frac{1}{n}\sum_{i=1}^{n} x_i + (1-g)\max_i x_i$$

where $g = \sigma(w_1 x_1 + \ldots + w_n x_n)$ with w_i trainable weights.

Ordinal Pooling was introduced by Deliège et al. [2]. The values within the window are sorted, and then reduced to one value using a trainable convex combination,

$$OP(\vec{x}) = \sum_{i=1}^{n} w_i x_{\pi_{x_1,\ldots,x_n}}(i)$$

where $\pi_{x_1,\ldots,x_n} : \{1,\ldots,n\} \to \{1,\ldots,n\}$ is a permutation that sorts the values x_i in ascending order.

Learned Norm Pooling was introduced by Gulcehre et al. [3]. They proposed to use the ℓ^p-norm with a trainable parameter $p > 1$, which can be formulated as

$$LNP(\vec{x}) = \sqrt[p]{\frac{1}{n}\sum_{i=1}^{n} |x_i|^p}.$$

If the input $\vec{x}$ consists of just positive values, then average- as well as max-pooling can be considered as applying the 1- or ∞-norm (up to some constant factor), both of which are special cases of the well known ℓ^p-norms.

Smooth Maximum Pooling

We proposed a new method that also generalizes average- and max-pooling, based on a smooth approximation of the maximum.

It includes a parameter $\tau \in \mathbb{R}$.

$$SMP_\tau(\vec{x}) := \sum_{i=1}^{n} x_i \frac{e^{\tau x_i}}{\sum_{i=1}^{n} e^{\tau x_i}}$$

It is again a convex combination of the arguments and includes

- average-pooling for $\tau = 0$,
- max-pooling for $\tau \to \infty$,
- min-pooling for $\tau \to -\infty$.

We investigated three variations:

- A fixed τ, initialized as $\tau_c = \log(c/C)$ for every channel $c = 1, \ldots, C$. (`SMPF`)
- A trainable τ, initially drawn from $N(0,1)$. (`SMP`)
- Using a separate trainable network branch to compute values of τ. (`SESMP`)

Results

We compared the methods by applying them to a VGG16 classification network which we trained on natural images of 200 classes of the Google Open Images v5 dataset.

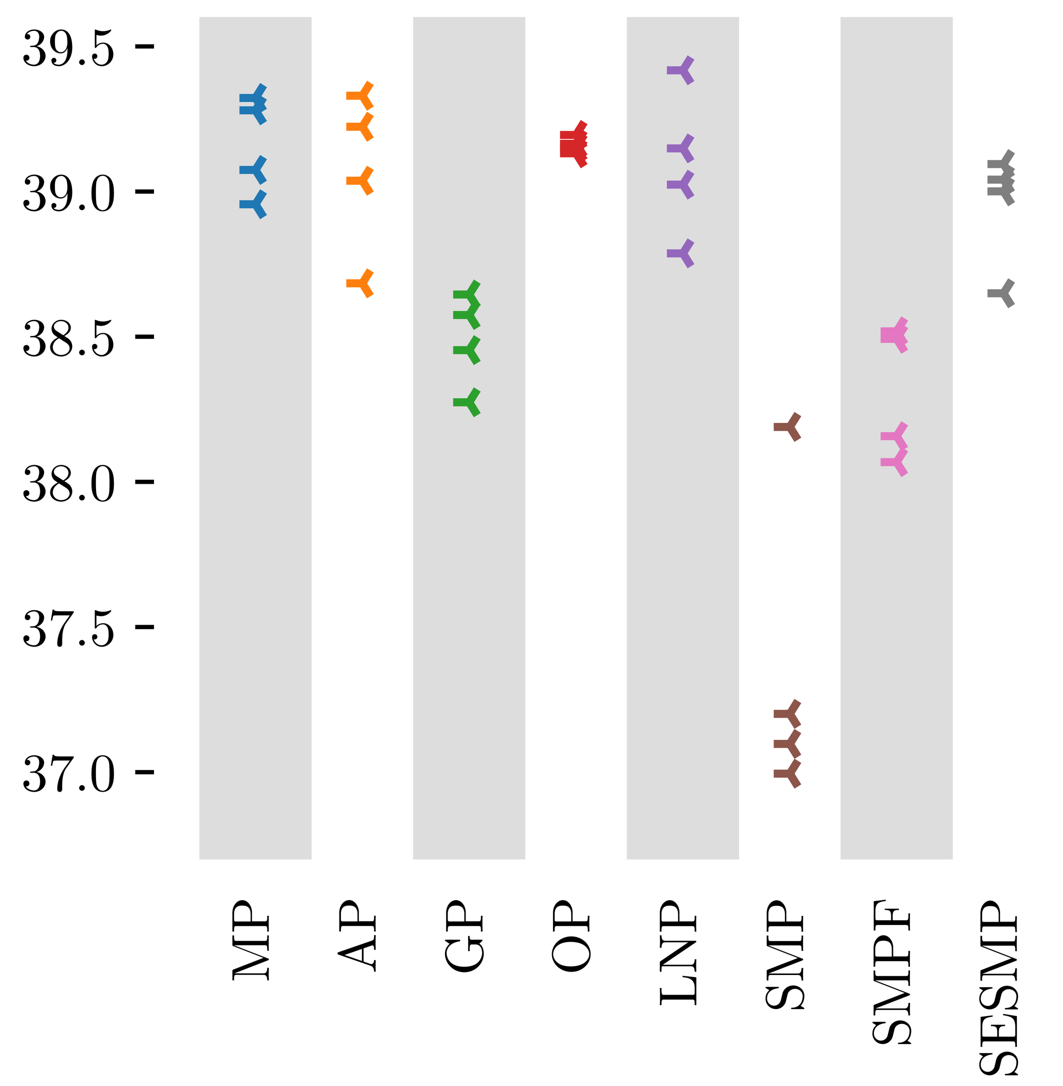

Fig. 2: Accuracy of each of the four runs per method on the test set.

For this classification task, none of the methods we investigated significantly outperformed the popular average- and max-pooling. We therefore concluded that in this setting more sophisticated pooling methods do not offer any advantage. They should be avoided due to their additional complexity which result in higher memory usage and higher computational cost.

References

[1] Lee et al., "Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree" *PMLR*, 2015

[2] Deliège et al. "Ordinal pooling" *BMVC*, 2019

[3] Gulcehre et al. "Learned norm pooling for deep feedforward and recurrent neural networks" *ECML*, 2014

[4] Simonyan et al. "Very deep convolutional networks for large-scale image recognition" *ICLR*, 2015