METHOD

Overall our method is a conceptually simple: take a convolutional layer and decompose its kernel using CP-decomposition.

The most “important” and time-consuming operation within modern CNNs is the generalized con- volution that maps an input tensor U (·, ·, ·) of size X ×Y ×S into an output tensor V (·, ·, ·) of size (X–d+1)×(Y –d+1)×T using the following linear mapping:

$$V(x,y,t)=\sum_{i=x-\delta}^{x+\delta}\sum_{j=y-\delta}^{y+\delta}\sum_{s=1}^{S}K(i-x+\delta,j-y+\delta,s,t)\,U(i,j,s)$$

Here, K(·, ·, ·, ·) is a 4D kernel tensor of size d×d×S×T with the first two dimensions correspond- ing to the spatial dimensions, the third dimension corresponding to different input channels, the fourth dimension corresponding to different output channels. The spatial width and height of the kernel are denoted as d, while δ denotes “half-width” (d – 1)/2 (for simplicity we assume square shaped kernels and even d).

The rank-R CP-decomposition (2) of the 4D kernel tensor has the form:

$$K(i,j,s,t)=\sum_{r=1}^{R}K^{x}(i-x+\delta,r)\,K^{y}(j-y+\delta,r)\,K^{s}(s,r)\,K^{t}(t,r)\,,$$

where $K^x(\cdot,\cdot)$, $K^y(\cdot,\cdot)$, $K^s(\cdot,\cdot)$, $K^t(\cdot,\cdot)$ are the four components of the composition representing

2D tensors (matrices) of sizes d×R, d×R, S×R, and T ×R respectively.

$$V(x,y,t)=\sum_{r=1}^{R}K^{t}(t,r)\left(\sum_{i=x-\delta}^{x+\delta}K^{x}(i-x+\delta,r)\left(\sum_{j=y-\delta}^{y+\delta}K^{y}(j-y+\delta,r)\left(\sum_{s=1}^{S}K^{s}(s,r)\,U(i,j,s)\right)\right)\right)$$

The output tensor V (·, ·, ·) can be computed from the input tensor U (·, ·, ·) via a sequence of four convolutions with smaller kernels.

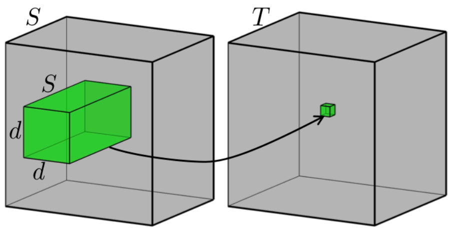
$$U^s(i,j,r) = \sum_{s=1}^{S} K^s(s,r)\, U(i,j,s)$$

$$U^{sy}(i,y,r) = \sum_{j=y-\delta}^{y+\delta} K^y(j-y+\delta,r)\, U^s(i,j,r)$$

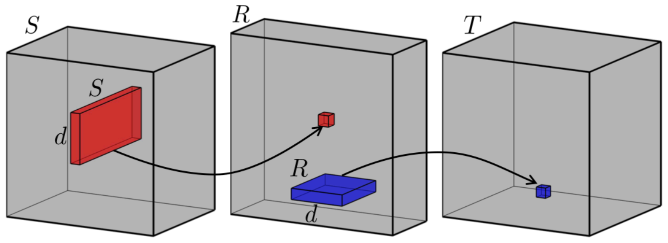
$$U^{syx}(x,y,r) = \sum_{i=x-\delta}^{x+\delta} K^x(i-x+\delta,r)\, U^{sy}(i,y,r)$$

$$V(x,y,t) = \sum_{r=1}^{R} K^t(t,r)\, U^{syx}(x,y,r)\,,$$

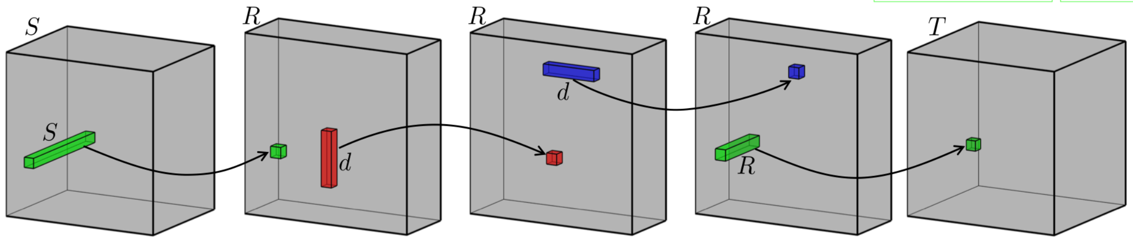
Computing $U^s(\cdot,\cdot,\cdot)$ from $U(\cdot,\cdot,\cdot)$ in as well as $V(\cdot,\cdot,\cdot)$ from $U^{syx}(\cdot,\cdot,\cdot)$ in represent so-called 1×1 convolutions (also used within "network-in-network" approach that essentially perform pixel-wise linear re-combination of input maps. Computing $U^{sy}(\cdot,\cdot,\cdot)$ from $U^s(\cdot,\cdot,\cdot)$ and $U^{syx}(\cdot,\cdot,\cdot)$ from $U^{sy}(\cdot,\cdot,\cdot)$ in and are "standard" convolutions with small kernels that are "flat" in one of the two spatial dimensions.



(a) Full convolution



(b) Two-component decomposition (Jaderberg et al., 2014a)



(c) CP-decomposition