Spectral Representations for Convolutional Neural Networks: Supplementary Material

1 Appendix A: Algorithmic Implementation Details

Here we provide additional detail pertaining to the specific algorithmic implementation of the spectral pooling and spectral parameterization. Algorithms 1 and 2 detail the steps required to compute the spectral pooling and corresponding back-propagation respectively. CROPSPECTRUM and PADSPECTRUM are self-explanatory: they crop or zero-pad the frequency spectrum to the appropriate dimensionalities, respectively.

Algorithm 1: Spectral pooling

Input: Map $\mathbf{x} \in \mathbb{R}^{M \times N}$, output size $H \times W$
Output: Pooled map $\hat{\mathbf{x}} \in \mathbb{R}^{H \times W}$

$\mathbf{y} \leftarrow \mathscr{F}(\mathbf{x})$
$\hat{\mathbf{y}} \leftarrow \text{CROPSPECTRUM}(\mathbf{y}, H \times W)$
$\hat{\mathbf{y}} \leftarrow \text{TREATCORNERCASES}(\hat{\mathbf{y}})$
$\hat{\mathbf{x}} \leftarrow \mathscr{F}^{-1}(\hat{\mathbf{y}})$

Algorithm 2: Spectral pooling back-propagation

Input: Gradient w.r.t output $\frac{\partial R}{\partial \hat{\mathbf{x}}}$
Output: Gradient w.r.t input $\frac{\partial R}{\partial \mathbf{x}}$

$\hat{\mathbf{z}} \leftarrow \mathscr{F}\left(\frac{\partial R}{\partial \hat{\mathbf{x}}}\right)$
$\hat{\mathbf{z}} \leftarrow \text{REMOVEREDUNDANCY}(\hat{\mathbf{z}})$
$\mathbf{z} \leftarrow \text{PADSPECTRUM}(\hat{\mathbf{z}}, M \times N)$
$\mathbf{z} \leftarrow \text{RECOVERMAP}(\mathbf{z})$
$\frac{\partial R}{\partial \mathbf{x}} \leftarrow \mathscr{F}^{-1}(\mathbf{z})$

Algorithm 3: TREATCORNERCASES

Input: Input map $\mathbf{y} \in \mathbb{C}^{M \times N}$
Output: Output map $\mathbf{z}$ with corner cases obeying conjugate symmetry, special case indices S

$\mathbf{z} \leftarrow \mathbf{y}$
$S \leftarrow \{(0,0)\}$
if M is even **then**
    $S \leftarrow \{(\frac{M}{2}, 0)\}$
end if
if N is even **then**
    $S \leftarrow \{(0, \frac{N}{2})\}$
end if
if M is even and N is even **then**
    $S \leftarrow \{(\frac{M}{2}, \frac{N}{2})\}$
end if
for $i \in S$ **do**
    $\text{Im}(\text{z}_i) \leftarrow 0$
end for

Algorithm 4: REMOVEREDUNDANCY

Input: Input gradient map $\mathbf{y} \in \mathbb{C}^{M \times N}$
Output: Gradient $\mathbf{z}$ in terms of unconstrained parameters only

$\mathbf{z}, S \leftarrow$ TREATCORNERCASES$(\mathbf{y})$
$I \leftarrow \emptyset$
for $m = 0, \ldots, M-1$ **do**
  for $n = 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$ **do**
    if $(m, n) \notin S$ **then**
      if $(m, n) \notin I$ **then**
        $\mathrm{z}_{m,n} \leftarrow 2\mathrm{z}_{m,n}$
        $I \leftarrow I \cup \{(m, n), ((M-m) \bmod M, (N-n) \bmod N)\}$
      else
        $\mathrm{z}_{m,n} \leftarrow 0$
      end if
    end if
  end for
end for

Algorithm 5: RECOVERMAP

Input: Input gradient $\mathbf{y} \in \mathbb{C}^{M \times N}$ parametrized by unconstrained elements only
Output: Full gradient $\mathbf{z}$ with recovered redundancy

$\mathbf{z}, S \leftarrow$ TREATCORNERCASES$(\mathbf{y})$
$I \leftarrow \emptyset$
for $m = 0, \ldots, M-1$ **do**
  for $n = 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$ **do**
    if $(m, n) \notin S$ **then**
      if $(m, n) \notin I$ **then**
        $\mathrm{z}_{m,n} \leftarrow \frac{1}{2}\mathrm{z}_{m,n}$
        $\mathrm{z}_{(M-m) \bmod M, (N-n) \bmod N} \leftarrow \mathrm{z}_{m,n}$
        $I \leftarrow I \cup \{(m, n), ((M-m) \bmod M, (N-n) \bmod N)\}$
      else
        $\mathrm{z}_{m,n} \leftarrow 0$
      end if
    end if
  end for
end for