Deep TEN: Texture Encoding Network
Supplementary Material

Hang Zhang  Jia Xue  Kristin Dana
Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, USA
{zhang.hang,jia.xue}@rutgers.edu, kdana@ece.rutgers.edu

Encoding Layer Implementations

This supplementary material provides the explicit expression for the gradients of the loss $\ell$ with respect to (w.r.t) the layer input and the parameters for implementing Encoding Layer. The $L2$-normalization as a standard component is used outside the encoding layer.

Gradients w.r.t Input $X$ The encoder $E = \{e_1, \ldots, e_K\}$ can be viewed as $k$ independent sub-encoders. Therefore the gradients of the loss function $\ell$ w.r.t input descriptor $x_i$ can be accumulated $\frac{d\ell}{dx_i} = \sum_{k=1}^{K} \frac{d\ell}{d_{c_k}} \frac{d_{c_k}}{dx_i}$. According to the chain rule, the gradients of the encoder w.r.t the input is given by

$$\frac{de_k}{dx_i} = r_{ik} \frac{da_{ik}}{dx_i} + a_{ik} \frac{dr_{ik}}{dx_i},$$

where $a_{ik}$ and $r_{ik}$ are defined in Sec 2 of the main paper, $\frac{dr_{ik}}{dx_i} = 1$. Let $f_{ik} = e^{-s_k \|r_{ik}\|^2}$ and $h_1 = \sum_{m=1}^{K} f_{im}$, we can write $a_{ik} = \frac{f_{ik}}{h_1}$. The derivatives of the assigning weight w.r.t the input descriptor is

$$\frac{da_{ik}}{dx_i} = \frac{1}{h_1} \cdot \frac{f_{ik}}{(h_1)^2} \cdot \sum_{m=1}^{K} \frac{df_{im}}{dx_i},$$

where $\frac{df_{ik}}{dx_i} = -2s_k f_{ik} \cdot r_{ik}$. 

Gradients w.r.t Codewords $C$ The sub-encoder $e_k$ only depends on the codeword $c_k$. Therefore, the gradient of loss function w.r.t the codeword is given by $\frac{d\ell}{dc_k} = \frac{dr_{ik}}{dc_k} \frac{dr_{ik}}{dx_i}$. 

$$\frac{dc_k}{dc_k} = \sum_{i=1}^{N} (r_{ik} \frac{da_{ik}}{dc_k} + a_{ik} \frac{dr_{ik}}{dc_k}),$$

where $\frac{dr_{ik}}{dc_k} = -1$. Let $g_{ik} = \sum_{m \neq k} f_{im}$. According to the chain rule, the derivatives of assigning w.r.t the codewords can be written as

$$\frac{d_{a_{ik}}}{dc_k} = \frac{d_{a_{ik}}}{d_{f_{ik}}} \frac{d_{f_{ik}}}{dc_k} = \frac{2s_k f_{ik} g_{ik}}{(h_1)^2} \cdot r_{ik}.$$  (4)

Gradients w.r.t Smoothing Factors Similar to the codewords, the sub-encoder $e_k$ only depends on the $k$-th smoothing factor $s_k$. Then, the gradient of the loss function w.r.t the smoothing weight is given by $\frac{d\ell}{ds_k} = \frac{dr_{ik}}{ds_k} \frac{dr_{ik}}{dx_i}$. 

$$\frac{ds_k}{ds_k} = -\frac{f_{ik} g_{ik} \|r_{ik}\|^2}{(h_1)^2}.$$  (5)

Note In practice, we multiply the numerator and denominator of the assigning weight with $e^{\phi_i}$ to avoid overflow:

$$a_{ik} = \frac{\exp(-s_k \|r_{ik}\|^2 + \phi_i)}{\sum_{j=1}^{K} \exp(-s_j \|r_{ij}\|^2 + \phi_i)},$$  (6)

where $\phi_i = \min_k \{s_k \|r_{ik}\|^2\}$. Then $\frac{d\phi_k}{ds_k} = e^{\phi_i} f_{ik}$. 

A Torch implementation is provided in supplementary material and available at https://github.com/zhanghang1989/Deep-Encoding