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 ℓ 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 $E = \{e_1, ...e_K\}$ can be viewed as k independent sub-encoders. Therefore the gradients of the loss function ℓ *w.r.t* input descriptor x_i can be accumulated $\frac{d_\ell}{d_{x_i}} = \sum_{k=1}^{K} \frac{d_\ell}{d_{e_k}} \cdot \frac{d_{e_k}}{d_{x_i}}$. According to the chain rule, the gradients of the encoder $w.r.t$ the input is given by

$$\frac{d_{e_k}}{d_{x_i}} = r_{ik}^T \frac{d_{a_{ik}}}{d_{x_i}} + a_{ik} \frac{d_{r_{ik}}}{d_{x_i}}, \tag{1}$$

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

$$\frac{d_{a_{ik}}}{d_{x_i}} = \frac{1}{h_i} \cdot \frac{d_{f_{ik}}}{d_{x_i}} - \frac{f_{ik}}{(h_i)^2} \cdot \sum_{m=1}^{K} \frac{d_{f_{im}}}{d_{x_i}}, \tag{2}$$

where $\frac{d_{f_{ik}}}{d_{x_i}} = -2 s_k f_{ik} \cdot r_{ik}$.

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 $\frac{d_\ell}{d_{c_k}} = \frac{d_\ell}{d_{e_k}} \cdot \frac{d_{e_k}}{d_{c_k}}$.

$$\frac{d_{e_k}}{d_{c_k}} = \sum_{i=1}^{N} (r_{ik}^T \frac{d_{a_{ik}}}{d_{c_k}} + a_{ik} \frac{d_{r_{ik}}}{d_{c_k}}), \tag{3}$$

where $\frac{d_{r_{ik}}}{d_{c_k}} = -1$. 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

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

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 $\frac{d_\ell}{d_{s_k}} = \frac{d_\ell}{d_{e_k}} \cdot \frac{d_{e_k}}{d_{s_k}}$.

$$\frac{d_{e_k}}{d_{s_k}} = -\frac{f_{ik} g_{ik} \|r_{ik}\|^2}{(h_i)^2} \tag{5}$$

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

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

where $\phi_i = \min_k \{s_k \|r_{ik}\|^2\}$. Then $\frac{d_{\bar{f}_{ik}}}{d_{x_i}} = e^{\phi_i} \frac{f_{ik}}{d_{x_i}}$.

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