Supplementary Material - Leveraging Seen and Unseen Semantic Relationships for Generative Zero-Shot Learning

Maunil R Vyas, Hemanth Venkateswara, and Sethuraman Panchanathan

Arizona State University, Tempe AZ 85281, USA
{mrvyas, hemanthv, panch}@asu.edu

1 Time complexity of the SR-loss

1. **For the semantic similarity:**
   - Cosine Similarity matrix $(n \times n)$: $\mathcal{O}(n^2 L)$, L length of semantic feature, n total classes
   - To get the K similar classes : $\mathcal{O}(n^2 \log K)$, $\mathcal{O}(n \log K)$ for each class, heap sort.
   - Overall cost : $\mathcal{O}(n^2 L) + \mathcal{O}(n^2 \log K)$
2. **For the visual similarity:**
   - Cosine Similarity matrix $(B \times K)$: $\mathcal{O}(BKV)$, V length of visual feature, B batch size classes, K neighbour classes.
   - Overall cost : $\mathcal{O}(BKVE)$, E total number of epochs.

- Time Complexity SR-loss (Clean Attributes): $\mathcal{O}(BKVE) + \mathcal{O}(n^2 L) + \mathcal{O}(n^2 \log K)$
- Time Complexity SR-loss (Noisy Text): $\mathcal{O}(BKVE) + \mathcal{O}(n^2 LE) + \mathcal{O}(n^2 \log KE)$

Clearly, the overall time complexity is linear in terms of E, K , B, V, and L and degree 2 polynomial with *log* in terms of the total classes. It is worth noticing that the complexity is not exponential and the running time cost is manageable.

2 Training Algorithm

Below, we illustrate our training procedure for the LsrGAN model. We train the Generator (G_θ) and Discriminator (D_θ) alternately with the Adam optimizer. Notice that the training of LsrGAN contains two phases, one for the seen classes and another for the unseen classes.

Algorithm 1 Training procedure for the LsrGAN

Input: number of epochs N_E, the batch size m, discriminator iterations $N_d = 5$ for seen classes, loss hyper parameters λ_c, λ_{vp} and λ_{sr}, $N_c = 1$ or 2 discriminator iterations for unseen classes, and Adam parameters $\beta_1 = 0.5$ and $\beta_2 = 0.9$.
for iter $= 1, ..., N_E$ **do**
    // Seen Class Training
    for $i = 1$, ..., N_d **do**
        Minibatch sampling from $\mathcal{T}^s$ with matching images from $\mathcal{X}^s$ and noise $\mathcal{Z}$
        $\tilde{\boldsymbol{x}} \leftarrow G_{\theta_g}(\boldsymbol{t}^s, \mathcal{Z})$
        Discriminator and classifier loss computation $\mathcal{L}_d$ and $\mathcal{L}_c$ using Eq. 2 and 3
        $\theta_d \leftarrow \text{Adam}(\nabla_{\theta_d^r}\mathcal{L}_d, \triangledown_{\theta_d^c}\mathcal{L}_c, \theta_d, \lambda_c, \beta_1, \beta_2)$
    end for
    Minibatch sampling from $\mathcal{T}^s$ and noise $\mathcal{Z}$
    Generator loss computation L_G using Eq. 8
    $\tilde{\boldsymbol{x}} \leftarrow G_{\theta_g}(\boldsymbol{t}^s, \mathcal{Z})$
    $\theta_g \leftarrow \text{Adam}(\nabla_{\theta_g}\mathcal{L}_d, \triangledown_{\theta_g}\mathcal{L}_{vp}, \triangledown_{\theta_g}\mathcal{L}_c, \triangledown_{\theta_g}\mathcal{L}_{sr}^s, \theta_g, \lambda_c, \lambda_{vp}, \lambda_{sr}, \beta_1, \beta_2)$
    // Unseen Class Training
    for $i = 1$, ..., N_c **do**
        Minibatch sampling from $\mathcal{T}^u$ and noise $\mathcal{Z}$
        $\tilde{\boldsymbol{x}} \leftarrow G_{\theta_g}(\boldsymbol{t}^u, \mathcal{Z})$
        Classifier loss computation $\mathcal{L}_c$ using Eq. 3
        $\theta_d^c \leftarrow \text{Adam}(\nabla_{\theta_d^c}\mathcal{L}_c, \theta_d^c, \lambda_c, \beta_1, \beta_2)$
    end for
    Minibatch sampling from $\mathcal{T}^u$ and noise $\mathcal{Z}$
    Generator loss computation L_G using Eq. 8
    $\tilde{\boldsymbol{x}} \leftarrow G_{\theta_g}(\boldsymbol{t}^u, \mathcal{Z})$
    $\theta_g \leftarrow \text{Adam}(\nabla_{\theta_g}\mathcal{L}_c, \triangledown_{\theta_g}\mathcal{L}_{sr}^u, \theta_g, \lambda_c, \lambda_{sr}, \beta_1, \beta_2)$
end for