Motivation

For real-time generation tasks, different devices may require models of different sizes due to their different computing power. And even for the same device, it also needs models of different sizes due to different power modes. Consequently, numerous models need to be trained and deployed for a single task.

To combat the above issues, we develop a width-switchable generator that can flexibly switch the width (the number of channels in layers) of the generator at runtime to accommodate various quality-efficiency trade-offs.

Methods

• To train the slimmable generator, we utilize adversarial training with multiple discriminators that share first several layers, and employ the Hinge loss:

\[
\max_G \mathbb{E}_{z \sim \mathcal{P}} [\min (0, -1 - D_{\text{w}}(G_{\text{w}}(z)))]
\]

\[
\min_D \mathbb{E}_{z \sim \mathcal{P}} [D_{\text{w}}(G_{\text{w}}(z))], \quad w = 1, 2, \ldots, W
\]

• To achieve consistency between these sub-generators, we propose stepwise in-place distillation that encourage narrower generators learn from wider ones:

\[
\min_G \max_L \sum_{i=1}^{W-1} \mathbb{E}_{z \sim \mathcal{P}} \left[ \left| \frac{G_{\text{w}}(z) - \Phi_G(G_{\text{w}}(z))}{\Phi_G(G_{\text{w}}(z))} \right|_2^2 \right]
\]

• To incorporate class-conditional label information into generators under the width-switchable mechanism, we present sliceable conditional batch normalization:

\[
x_{w,c_i} = \gamma_{w,c_i} \frac{x_{w,c_i} - \mu(x_{w,c_i})}{\sigma(x_{w,c_i})} + \beta_{c_i}
\]

Specifically, we slice cBN vectors \(\gamma_{c_i}\) and \(\beta_{c_i}\) to sub-vectors \(\gamma^s_{c_i}\) and \(\beta^s_{c_i}\) with the first \(s_i\) elements to match the length of sBN vectors \(\gamma_{w,c_i}\) and \(\beta_{w,c_i}\) at current width \(w\).

Evaluation Metrics

• FID and IS: evaluate the quality and diversity of generated samples.

• mIC: evaluate the consistency between generators at different widths, and is defined as follows:

\[
mIC = \frac{1}{N(N-1)} \sum_{w=1}^{W} \sum_{w'=1}^{W} \mathbb{E}_{z \sim \mathcal{P}} \left[ \Phi(G_{w}(z)) - \Phi(G_{w'}(z)) \right]_2^2
\]

where \(\Phi(\cdot)\) extracts the image features from the Inception v3 network.