Appendix for "Global versus Localized Generative Adversarial Nets"

Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki
guojun.qi@ucf.edu, {lihengzhang1993,hao_hu,m.edraki}@knights.ucf.edu
Jingdong Wang, and Xian-Sheng Hua
welleast@outlook.com, xiansheng.hxs@alibaba-inc.com

A. More Results on Semi-supervised Learning

This section describes more experiment results on the semi-supervised learning using LGAN. Besides the small discriminator structure employed in section 5.3, we further test LGAN with a larger CNN architecture, which is the same one as the "Conv-Large" used in [2]. For convenience, we will refer this larger discriminator as Conv-Large, while the one used in Section 5.3 as "Conv-Small" in the following. We compare the Conv-Large LGAN with the state-of-the-art semi-supervised learning methods (which are not necessarily the GAN-based) and report the results on both CIFAR-10 and CIFAR-100 datasets in this section. The architecture of the generator will keep the same as that used in Conv-Small experiments.

A.1. Discriminator Architectures

Table 1 and 2 summarize the architecture of Conv-Small and Conv-Large, respectively. We also apply the weight normalization [5] to all convolutional and dense layers in both architectures.

Name	Description
Input	32×32 RGB image
drop1	Dropout $p = 0.2$
conv1a	96, 3×3, pad=1, stride=1, LReLU
conv1b	96, 3×3, pad=1, stride=1, LReLU
conv1c	96, 3×3, pad=1, stride=2, LReLU
drop2	Dropout $p = 0.5$
conv2a	192, 3×3, pad=1, stride=1, LReLU
conv2b	192, 3×3, pad=1, stride=1, LReLU
conv2c	192, 3×3, pad=1, stride=2, LReLU
drop3	Dropout $p = 0.5$
conv3a	192, 3×3, pad=0, stride=1, LReLU
conv3b	192, 1×1, LReLU
conv3c	192, 1×1, LReLU
pool1	Global mean pooling $6 \times 6 \to 1 \times 1$
dense	Fully connected $192 \to 10$
output	Softmax

Table 1. The network architectures of Conv-Small

A.2. Training Details for Conv-Large

Like in training the Conv-Small, we adopt Adam optimizer to train both the discriminator and generator. The learning rate is set to 4×10^{-4}, and the maximal training epoch is $1,200$. We gradually anneal the learning rates to zero during the last 400 epochs. The other settings are kept as same as those for training Conv-Small (Section 5.3). For CIFAR-100, which consists of $50,000$ 32×32 training images and $10,000$ test images in a hundred classes, we change the dropout rate of drop1 layer from 0.2 to 0.1 and the output dimension of the last layer to 100. We also adopt early stopping – the training is terminated if the validation error stops decreasing over 100 consecutive epochs after the

Name	Description
Input	32×32 RGB image
drop1	Dropout $p = 0.2$
conv1a	128, 3×3, pad=1, stride=1, LReLU
conv1b	128, 3×3, pad=1, stride=1, LReLU
conv1c	128, 3×3, pad=1, stride=1, LReLU
pool1	Maxpooling 2×2
drop2	Dropout $p = 0.5$
conv2a	256, 3×3, pad=1, stride=1, LReLU
conv2b	256, 3×3, pad=1, stride=1, LReLU
conv2c	256, 3×3, pad=1, stride=1, LReLU
pool2	Maxpooling 2×2
drop3	Dropout $p = 0.5$
conv3a	512, 3×3, pad=0, stride=1, LReLU
conv3b	256, 1×1, LReLU
conv3c	128, 1×1, LReLU
pool3	Global mean pooling $6 \times 6 \to 1 \times 1$
drop4	Dropout $p = 0.1$
dense	Fully connected $128 \to 10$
output	Softmax

Table 2. The network architectures of Conv-Large

Method	CIFAR-10	CIFAR-100
Π model [1]	12.36 ± 0.31	39.19 ± 0.36
Temporal Ensembling [1]	12.16 ± 0.24	38.65 ± 0.51
Sajjadi et al. [4]	11.29 ± 0.24	-
VAT [2]	10.55	-
VadD [3]	11.32 ± 0.11	-
LGAN (Conv-Large)	$\mathbf{9.77 \pm 0.13}$	$\mathbf{35.52 \pm 0.33}$

Table 3. Classification errors on both CIFAR-10 and CIFAR-100 with $4,000$ and $10,000$ labeled training examples respectively. The best result is highlighted in bold.

600th epoch. The two hyper-parameters μ and η are chosen based on a separate validation set.

A.3. Experimental Results for Conv-Large

We compare the LGAN using Conv-Large discriminator with state-of-the-art semi-supervised baselines. The results are reported in Table 3. Note that we used $4,000$ and $10,000$ labeled training examples for CIFAR-10 (400 images per class) and CIFAR-100 (100 images per class) respectively and the rest of training data unlabeled. From the table, we can see that LGAN with Conv-Large outperforms the other compared methods on both datasets.

References

[1] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 2

[2] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arXiv preprint arXiv:1704.03976*, 2017. 1, 2

[3] S. Park, J.-K. Park, S.-J. Shin, and I.-C. Moon. Adversarial dropout for supervised and semi-supervised learning. *arXiv preprint arXiv:1707.03631*, 2017. 2

[4] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016. 2

[5] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016. 1