Supplementary Materials for Tangent-Normal Adversarial Regularization for Semi-supervised Learning

Bing Yu[1]*, Jingfeng Wu[1]*, Jinwen Ma[1], Zhanxing Zhu[123]†
[1]School of Mathematical Sciences, Peking University
[2]Center for Data Science, Peking University
[3] Beijing Institute of Big Data Research
{byu,pkuwjf}@pku.edu.cn jwma@math.pku.edu.cn zhanxing.zhu@pku.edu.cn

1. Two-rings dataset

The underlying manifold for two-rings data is given by $\mathcal{M} = \mathcal{M}_+ \cup \mathcal{M}_-$, where

$$\mathcal{M}_+ = \left\{ (x_1, x_2) \,\middle|\, x_1^2 + x_2^2 = 0.9^2 \right\}$$
$$\mathcal{M}_- = \left\{ (x_1, x_2) \,\middle|\, x_1^2 + x_2^2 = 1.1^2 \right\}.$$

The observed data is sampled as $x = x_0 + n$, where x_0 is uniformly sampled from $\mathcal{M}$ and $n \sim \mathcal{N}(0, 2^{-2})$. We sample 6 labeled training data, 3 for each class, and $3,000$ unlabeled training data, as shown in Figure 3.

2. Experiments details on FashionMNIST

In FashionMNIST[1] experiments, we preserve 100 data points for validation from the original training dataset. That is, we use $100/200/1,000$ labeled data for training and another 100 labeled data for validation. For pre-processing, we scale pixel values of images into $[0, 1]$. The architecture of the neural network for classification is as following: (a, b) denotes the convolution filter is with $a \times a$ shape and b channels. The max pooling layer is with stride 2. And we apply local response normalization (LRN) [4]. The number of hidden nodes in the first fully connected layer is 512. And the number of hidden nodes in the last fully connected layer is 10.

$$\text{Conv}(3, 32) \to \text{ReLU} \to \text{Conv}(3, 32) \to \text{ReLU} \to \text{MaxPooling} \to \text{LRN} \to \text{Conv}(3, 64) \to \text{ReLU} \to \text{Conv}(3, 64) \to \text{ReLU} \to \text{MaxPooling} \to \text{LRN} \to \text{FC1} \to \text{ReLU} \to \text{FC2}$$

For the labeled data, the batch size is 32, and for the unlabeled data, the batch size is 128. All networks are trained for $12,000$ updates. The optimizer is ADAM with initial learning rate 0.001, and linearly decayed over the last $4,000$ updates. The hyperparameters tuned include 1) the magnitude of the tangent adversarial perturbation ϵ_1, 2) the magnitude of the normal adversarial perturbation ϵ_2 and 3) the hyperparameter λ in Eq. (28). All other hyperparameters are set as 1.0 (e.g., $\alpha_1, \alpha_2, \alpha_3 = 1.0$). We tune λ from $\{1, 0.1, 0.01, 0.001\}$, and ϵ_1, ϵ_2 randomly from $[0.05, 20]$. The corresponding hyperparameters used for experiments in the main paper are reported in Table 1.

The encoder of the VAE for identifying the underlying manifold is a LeNet-like one, with two convolutional layers and one fully connected layer. And the decoder is symmetric with the encoder, except using deconvolutional layers to replace convolutional ones. The latent dimensionality is 128. The localized GAN for identifying the underlying manifold is similar as stated in [3]. And the implementation is modified from https://github.com/z331565360/Localized-GAN. We change the latent dimensionality into 128.

VAE is pretrained and fixed during the training of TNAR. We tried both jointly and separately training the LGAN with the classifier, observing no significant difference.

3. Experiments details on SVHN and CIFAR-10

In SVHN[2] and CIFAR-10[3] experiments, we preserve $1,000$ data for validation from the original training set. That is, we use $1,000/4,000$ labeled data for training and another $1,000$ labeled data for validation. The only preprocessing on data is to scale the pixels value into $[0, 1]$. We do not use data augmentation. The structure of classification neural network is shown in Table 2, which is identical

*Equal contributions.
†Corresponding author.
[1]https://github.com/zalandoresearch/fashion-mnist
[2]http://ufldl.stanford.edu/housenumbers/
[3]https://www.cs.toronto.edu/~kriz/cifar.html

Table 1. The hyperparameters for TNAR on FashionMNIST dataset. λ: the hyperparameter in Eq. (28);ϵ_1 and ϵ_2: the norms of tangent adversarial perturbation and normal adversarial perturbation.

Hyperparameters	100 labels	200 labels	1000 labels
$\lambda/\epsilon_1/\epsilon_2$ for TNAR-LGAN	0.1/0.2/8.0	0.1/4.0/0.5	1.0/20.0/0.5
$\lambda/\epsilon_1/\epsilon_2$ for TNAR-VAE	1.0/2.0/0.05	1.0/0.5/0.05	1.0/5.0/6.0

as in [2].

For the labeled data, the batch size is 32, and for the unlabeled data, the batch size is 128. For SVHN, all networks are trained for $48,000$ updates. And for CIFAR-10, all networks are trained for $200,000$ updates. The optimizer is ADAM with initial learning rate 0.001, and linearly decayed over the last $16,000$ updates. The hyperparameters tuned include 1) the magnitude of the tangent adversarial perturbation ϵ_1, 2) the magnitude of the normal adversarial perturbation ϵ_2 and 3) the hyperparameter λ in Eq. (28). All other hyperparameters are set as 1.0 (e.g., $\alpha_1, \alpha_2, \alpha_3 = 1.0$). We tune λ from $\{1, 0.1, 0.01, 0.001\}$, and ϵ_1, ϵ_2 randomly from $[0.05, 20]$.

The VAE for identifying the underlying manifold for SVHN and CIFAR-10 is implemented as in https://github.com/axium/VAE-SVHN. The only modification is we change the coefficient of the regularization term from 0.01 to 1. The localized GAN for learning the underlying manifold for SVHN and CIFAR-10 is similar as stated in [3]. And the implementation is modified from https://github.com/z331565360/Localized-GAN. We change the latent dimensionality of VAE and localized GAN into 512 for both SVHN and CIFAR-10. The hyperparameters used in the main paper for TNAR are reported in Table 3.

4. More adversarial examples

More adversarial perturbations and adversarial examples in the tangent space and normal space are shown in Figure 1 and Figure 2.

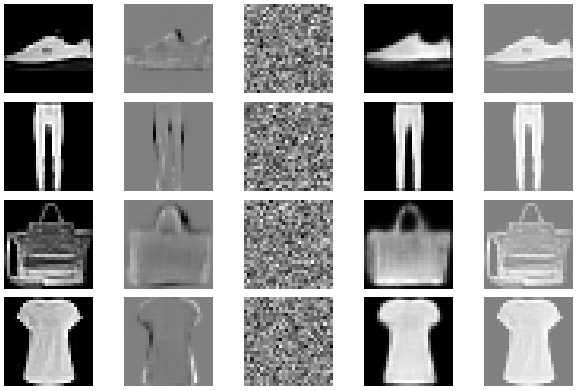

Figure 1. The perturbations and adversarial examples in the tangent space and normal space for FashionMNIST dataset. Note that since the perturbations are actually too small, to distinguish them visually, thus we show the scaled perturbations. From left to right: original example, tangent adversarial perturbation, normal adversarial perturbation, tangent adversarial example, normal adversarial example.

Figure 2. The perturbations and adversarial examples in tangent space and normal space for CIFAR-10 dataset. Note that the perturbations is actually too small to distinguish easily, thus we show the scaled perturbations. From left to right: original example, tangent adversarial perturbation, normal adversarial perturbation, tangent adversarial example, normal adversarial example.

References

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

[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.

[3] G.-J. Qi, L. Zhang, H. Hu, M. Edraki, J. Wang, and X.-S. Hua. Global versus localized generative adversarial nets. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] A. E. Robinson, P. S. Hammon, and V. R. de Sa. Explaining brightness illusions using spatial filtering and local response normalization. *Vision research*, 47(12):1631–1644, 2007.

[5] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[6] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Table 2. The structure of convolutional neural networks for experiments on CIFAR-10 and SVHN, based on [6, 5, 1]. All the convolutional layers and fully connected layers are followed by batch normalization except the fully connected layer on CIFAR-10. The slopes of all lReLU functions in the networks are 0.1.

Conv-Small on SVHN	Conv-Small on CIFAR-10	Conv-Large
32×32 RGB image		
3×3 conv. 64 lReLU	3×3 conv. 96 lReLU	3×3 conv. 128 lReLU
3×3 conv. 64 lReLU	3×3 conv. 96 lReLU	3×3 conv. 128 lReLU
3×3 conv. 64 lReLU	3×3 conv. 96 lReLU	3×3 conv. 128 lReLU
2×2 max-pool, stride 2 dropout, $p = 0.5$		
3×3 conv. 128 lReLU	3×3 conv. 192 lReLU	3×3 conv. 256 lReLU
3×3 conv. 128 lReLU	3×3 conv. 192 lReLU	3×3 conv. 256 lReLU
3×3 conv. 128 lReLU	3×3 conv. 192 lReLU	3×3 conv. 256 lReLU
2×2 max-pool, stride 2 dropout, $p = 0.5$		
3×3 conv. 128 lReLU	3×3 conv. 192 lReLU	3×3 conv. 512 lReLU
1×1 conv. 128 lReLU	1×1 conv. 192 lReLU	1×1 conv. 256 lReLU
1×1 conv. 128 lReLU	1×1 conv. 192 lReLU	1×1 conv. 128 lReLU
global average pool, 6×6 → 1×1		
dense 128 → 10	dense 192→ 10	dense 128→ 10
10-way softmax		

Table 3. The hyperparameters for TNAR on SVHN and CIFAR-10 datasets. λ: the hyperparameter in Eq. (28);ϵ_1 and ϵ_2: the norms of tangent adversarial perturbation and normal adversarial perturbation.

Hyperparameters	SVHN 1,000 labels	CIFAR-10 4,000 labels	SVHN 1,000 labels with augmentation	CIFAR-10 4,000 labels with augmentation
$\lambda/\epsilon_1/\epsilon_2$ for TNAR-LGAN (small)	0.01/0.5/2.0	0.1/5.0/1.0	-	-
$\lambda/\epsilon_1/\epsilon_2$ for TNAR-LGAN (large)	0.01/0.5/2.0	0.1/5.0/1.0	-	-
$\lambda/\epsilon_1/\epsilon_2$ for TNAR-VAE (small)	0.01/0.2/2.0	0.01/5.0/1.0	-	-
$\lambda/\epsilon_1/\epsilon_2$ for TNAR-VAE (large)	0.01/0.2/2.0	0.001/5.0/1.0	0.01/0.2/2.0	0.001/4.0/1.0