Auto-encoder with Adversarially Regularized Latent Variables for Semi-Supervised Learning

Ryosuke Tachibana *, Takashi Matsubara *, Kuniaki Uehara *

Abstract

The amount of accessible raw data is ever-increasing in spite of the difficulty in obtaining a variety of labeled information; this makes semi-supervised learning a topic of practical importance. This paper proposes a novel regularization algorithm of an autoencoding deep neural network for semi-supervised learning. Given an input data, the deep neural network outputs the estimated label, and the remaining information called style. On the basis of the framework of a generative adversarial network, the proposed algorithm regularizes the label and the style according to a prior distribution, separating the label explicitly from the style. As a result, the deep neural network is trained to estimate correct labels by using a limitedly labeled dataset. The proposed algorithm achieved accuracy comparable with or superior to that of the existing state-of-the-art semi-supervised algorithms for the benchmark tasks, the MNIST database, and the SVHN dataset.

Keywords: Auto-encoder, Deep Learning, Generative Adversarial Networks, Semi-Supervised Learning

1 Introduction

While the amount of accessible data is ever-increasing, it often lacks auxiliary information such as class labels, and hand-labeling of the entire dataset is impossible or, at least, not economical. Therefore, there is considerable practical interest in semi-supervised learning. Semi-supervised learning deals with a classification task under the condition that only a small subset of a given dataset has corresponding class labels [1]. Semi-supervised algorithms utilize a large amount of unlabeled data and enable a more accurate classification than classifiers trained only with labeled data. Neural networks with deep architectures, also known as *deep learning*, performed impressively on a wide range of machine learning tasks, including semi-supervised learning [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Numerous approaches employed an unsupervised dimension reduction called *autoencoder* (AE) [4], consisting of two neural networks; encoder and decoder. The encoder is trained as a classifier in supervised learning, whereas the decoder is trained to reconstruct the given data, working as an additional penalty [5, 6, 7]. Some autoencoder-based approaches implemented unsupervised Bayesian inference and a generative procedure of a given dataset

* Graduate School of System Informatics, Kobe University, Hyogo, Japan

on an autoencoder, along with supervised classification [8, 9, 17]. They naturally require computation of the integral of the posterior distribution over latent variables or Monte Carlo sampling from it, thereby increasing their computational time. Other methods were based on the framework of the generative adversarial network (GAN) [10, 11, 12, 13, 14], obtaining latent representations of a given dataset by comparison with artificially generated datasets. The objective function of the GAN is an additional penalty for supervised classification. The learning procedure thereof is prone to destabilization, and thus requires careful adjustment of architectures and parameters [18, 19].

Therefore, finding a good penalty is the key to success in semi-supervised learning based on deep neural networks. Recently, a simple regularization approach for the autoencoder, called the adversarial autoencoder (AAE), was proposed for unsupervised dimension reduction [20]: It regularizes the latent representations using the framework of the GAN. This paper proposes a novel semi-supervised learning algorithm called *adversarial regularization* for autoencoder based on the principle of the adversarial autoencoder. Given a labeled data, the autoencoder is trained to estimate the correct label. Even without label information, the autoencoder outputs the estimated label, and a latent representation called *style*. The proposed algorithm regularizes the label and style according to a joint prior distribution on the basis of the framework of the GAN. As a result, a given data is decomposed explicitly into the label information contributing to the classification task, and the remaining style information. The proposed algorithm is evaluated using the semi-supervised tasks of the MNIST database and SVHN dataset as a benchmark. It achieves impressive accuracy, comparable or superior to the existing state-of-the-art semi-supervised algorithms based on deep neural networks. The preliminary and limited results are presented in a conference paper [21].

2 Related Works

While some semi-supervised learning algorithms focused on manifold learning [15, 16], many studies on deep neural networks employed a combination of unsupervised dimension reduction and supervised classification [1]. Previous studies employed autoencoder as an unsupervised dimension reduction. An autoencoder consists of two neural networks (encoder and decoder) and is trained by minimizing the objective function called the reconstruction error $\mathscr{L}_{rec}$ [5, 6, 7, 8, 9]:

$$\mathscr{L}_{rec}(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[L(\boldsymbol{x}, \hat{\boldsymbol{x}})\right], \tag{1}$$

where $p_{data}(\boldsymbol{x})$ denotes a given dataset; the encoder outputs the latent representation $\boldsymbol{z} = q(\boldsymbol{x}; \boldsymbol{\theta}_q)$ given an input; the decoder reconstructs the input $\hat{\boldsymbol{x}} = p(\boldsymbol{z}; \boldsymbol{\theta}_p)$; and $L(\cdot, \cdot)$ is a distance function, which is typically the mean squared error. Previous studies proposed a *variational autoencoder* (VAE), which is an implementation of unsupervised Bayesian inference and generative procedure of a given dataset on the encoder q and decoder p [17, 8, 9]. The objective function $\mathscr{L}_{vae}$ is derived according to variational methods, as follows:

$$\mathscr{L}_{vae}(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[D_{KL}(q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}_q)||p_{prior}(\boldsymbol{z})) - \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}_q)}[\log p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\theta}_p)]\right], \tag{2}$$

where D_{KL} is the Kullback-Leibler divergence and $p_{prior}(\boldsymbol{z})$ is the prior distribution of the latent representations $\boldsymbol{z}$. The first and second terms can be considered as regularization terms and a general form of the reconstruction error $\mathscr{L}_{rec}$. The framework of the GAN

provides a further unsupervised dimension reduction [10]. A GAN consists of two neural networks, generator G and discriminator D. Given a latent representation $\boldsymbol{z}$ randomly chosen from a prior distribution $p_{prior}(\boldsymbol{z})$, the generator G outputs an artificial data $\hat{\boldsymbol{x}}$. The discriminator D is trained to classify the real data $\boldsymbol{x}$ from the artificial data $\hat{\boldsymbol{x}}$, whereas the generator G is trained to "trick" the discriminator D as follows:

$$\min_{\theta_G} \max_{\theta_D} \mathscr{L}_{adv}(\theta_D, \theta_G), \tag{3}$$

where

$$\mathscr{L}_{adv}(\theta_D, \theta_G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x};\theta_D)] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z};\theta_G);\theta_D))] \tag{4}$$

The output of the discriminator D denotes the probability that the input is sampled from the given dataset $p_{data}(\boldsymbol{x})$ rather than from the generator G. At the end of the training, the discriminator D is expected to output 0.5 for the input from both the given dataset $p_{data}(\boldsymbol{x})$ and the generator G, and the generator G is expected to output an artificial dataset that is distributed as the distribution of the given dataset $\boldsymbol{x}$. When an alternative neural network is trained to estimate the latent representation $\boldsymbol{z}$ given an input artificial data $\hat{\boldsymbol{x}}$, the neural network and generator G work as the encoder p and decoder q, respectively [22]. The learning procedure is known to be prone to be destabilized because the distribution shape of the given dataset $\boldsymbol{x}$ is complicated and it is difficult to model it using the generator G. The GAN requires careful adjustment of its architecture and parameters [18, 19]. The framework of the GAN can be used for regularization of the latent representation $\boldsymbol{z}$ of the autoencoder; this is called the adversarial autoencoder [20]. In this case, the encoder q of the autoencoder corresponds to the generator G of the GAN, and is expected to output latent representations $q(\boldsymbol{z}|\boldsymbol{x};\theta_q)$ that are distributed as in the prior distribution $p_{prior}(\boldsymbol{z})$. Since the prior distribution $p(\boldsymbol{z})$ is far simpler than that of the given dataset $\boldsymbol{x}$, its learning procedure is easier and more robust.

For semi-supervised classification tasks, these unsupervised dimension reductions are combined with alternative supervised or semi-supervised learning algorithms. In some approaches, the latent representation $\boldsymbol{z}$ extracted from the input $\boldsymbol{x}$ by the encoder p is classified using another supervised or semi-supervised classification algorithm [8, 11, 14]. However, unsupervised dimension reduction is sometimes harmful to classification because it has a risk of extracting only information useless for classification, e.g., penmanship of handwritten strings in a classification of characters. Alternative approaches employ the objective function of the unsupervised dimension reduction as an additional penalty of supervised classification [5, 6, 7, 8, 9, 12, 13]. In this case, the encoder p of the autoencoder and the discriminator D of the GAN work as a classifier C and are trained with the following objective function

$$\mathscr{L}_{cls}(\theta_c) = \mathbb{E}_{\boldsymbol{x},y\sim p_{labeled}(\boldsymbol{x})}\left[-\sum_k \mathbb{I}(y=k)\log C(y|\boldsymbol{x};\theta_c)\right], \tag{5}$$

where y denotes the label corresponding to the input $\boldsymbol{x}$; $p_{labeled}(\boldsymbol{x})$ is the labeled subset of the given dataset $p_{data}(\boldsymbol{x})$; and $\mathbb{I}(cond)$ is the indicator function, which takes the value 1 when the condition $cond$ is satisfied and 0 otherwise. In these approaches, the latent representation $\boldsymbol{z}$ other than the label y is separated from the label y and is sometimes called *style*.

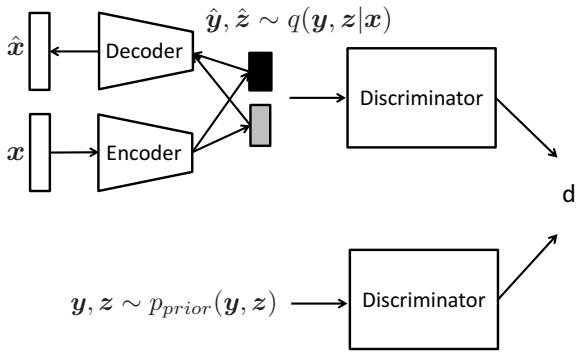

Figure 1: Diagram of the proposed autoencoder and adversarial regularization.

3 Adversarial Regularization for Semi-Supervised Learning

In this section, we propose a novel adversarial regularization algorithm of semi-supervised learning based on deep neural networks. First, we propose the autoencoder depicted in Fig. 1. Given an input $\boldsymbol{x}$, the encoder q, parameterized by θ_q, outputs two latent representations denoted $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$. Using the softmax activation function, the summation of the elements $\hat{y}_k$ of the latent representation $\hat{\boldsymbol{y}}$ is clamped to one, while the latent representation $\hat{\boldsymbol{z}}$ has no activation function, and thus its elements $\hat{z}_j$ are distributed over the real number. The decoder p, parameterized by θ_p, accepts the latent representations $\boldsymbol{y}$ and $\boldsymbol{z}$ and outputs an artificial data $\hat{\boldsymbol{x}}$ to reconstruct the input $\boldsymbol{x}$. As per ordinary autoencoders, encoder q and decoder p are trained by minimizing the reconstruction error

$$\mathscr{L}_{rec}(\theta_q, \theta_p) = \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})} \left[||\boldsymbol{x} - p(\hat{\boldsymbol{x}}|q(\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}|\boldsymbol{x}; \theta_q); \theta_p)||^2 \right]. \tag{6}$$

Note that, in spite of their formulations, the encoder q and the decoder p are deterministic in contrast to the variational autoencoder [17, 8, 9]. The encoder q works as a classifier and is trained by minimizing the classification error

$$\mathscr{L}_{cls}(\theta_q) = \mathbb{E}_{\boldsymbol{x}, y \sim p_{labeled}(\boldsymbol{x}, y)} \left[-\sum_k \mathbb{I}(y = k) \log q(\hat{y}_k|\boldsymbol{x}) \right]. \tag{7}$$

The latent representation $\hat{\boldsymbol{y}}$ denotes the posterior probability of the estimated label and the latent representation $\boldsymbol{z}$ represents the style of the input $\boldsymbol{x}$.

Here, we introduce the joint prior distribution $p_{prior}(\boldsymbol{y}, \boldsymbol{z})$ of the label $\hat{\boldsymbol{y}}$ and the style $\hat{\boldsymbol{z}}$. In contrast to the original study of the adversarial autoencoder, the label $\boldsymbol{y}$ and the style $\boldsymbol{z}$ are completely independent. The label $\boldsymbol{y}$ has only one element that takes one, with the remaining taking zero; this follows a uniform categorical distribution. Each element z_k of the style $\boldsymbol{z}$ follows a normal distribution with zero-mean and a variance of 5^2. Therefore,

$$\begin{aligned} p_{prior}(\boldsymbol{y}, \boldsymbol{z}) &= p_{prior}(\boldsymbol{y}) p_{prior}(\boldsymbol{z}) \\ &= \tfrac{1}{N_y} \textstyle\prod_k (2 \times 5^2 \pi)^{-\frac{1}{2}} \exp\left(-\dfrac{z_k^2}{2 \times 5^2} \right), \end{aligned} \tag{8}$$

where N_y denotes the number of the class label. Using the framework of the GAN, the encoder q is also trained to match the joint distribution of the latent representations $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$ to the joint prior distribution $p_{prior}(\boldsymbol{y}, \boldsymbol{z})$. The discriminator D, parameterized by θ_D,

Table 1: Results of semi-supervised classification.

	Test error (ave. (± std.) %)		
N_l	AE	AAE [20]	AR (ours)
100	3.41(±0.21)	1.90(±0.10)	0.98(±0.17)
300	2.54(±0.37)	—	0.97(±0.02)
1,000	1.98(±0.22)	1.60(±0.08)	0.75(±0.05)
3,000	1.57(±0.21)	—	0.70(±0.13)

Table 2: Comparison of error rates (%) between the adversarial regularization (AR) and other published results on the MNIST database, with 100 labels.

	Test error (ave. (± std.) %)
Methods	$N_l = 100$
DGM (2014) [8]	3.33(±0.14)
VAT (2015) [16]	2.12
CatGAN (2016) [12]	1.39(±0.28)
AAE (2015) [20]	1.90(±0.10)
ADGM(10MC) (2016) [9]	0.96(±0.02)
Improved-GAN (2016) [13]	0.96(±0.07)
Ladder (2015) [7]	0.86(±0.89)
AR (ours)	0.98(±0.17)

accepts the latent representations $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$ obtained from the encoder q or the samples $\boldsymbol{y}$ and $\boldsymbol{z}$ from the joint prior distribution $p_{prior}(\boldsymbol{y},\boldsymbol{z})$. The encoder q is trained by minimizing the adversarial regularization error

$$\begin{aligned}\mathscr{L}_{ar}(\theta_D,\theta_q) = \quad & \mathbb{E}_{\boldsymbol{y},\boldsymbol{z}\sim p(\boldsymbol{y},\boldsymbol{z})}[\log D(\boldsymbol{y},\boldsymbol{z};\theta_D)] \\ & +\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log\left(1-D(q(\hat{\boldsymbol{y}},\hat{\boldsymbol{z}}|\boldsymbol{x};\theta_q);\theta_D)\right)],\end{aligned} \tag{9}$$

while the discriminator D is trained by maximizing the adversarial regularization error $\mathscr{L}_{ar}$ Therefore, the objective function of the autoencoder with adversarial regularization is the weighted summation of all of the aforementioned errors:

$$\mathscr{L} = \lambda_{cls}\mathscr{L}_{cls} + \lambda_{rec}\mathscr{L}_{rec} + \lambda_{ar}\mathscr{L}_{ar}, \tag{10}$$

where λ_{cls}, λ_{rec}, and λ_{ar} are real-valued coefficients of the errors $\mathscr{L}_{cls}$, $\mathscr{L}_{rec}$, and $\mathscr{L}_{ar}$, respectively.

4 Results for Semi-Supervised Classification

4.1 Semi-Supervised Classification with Convolutional Neural Networks

The proposed algorithm was evaluated on the MNIST handwritten digit database [23] and the cropped version of the Street View House Numbers (SVHN) Dataset [24]. The MNIST database is a dataset of 28-by-28 grayscale images of 70,000 handwritten digits, comprising 60,000 images for training and 10,000 images for testing. We divided the 60,000 training images into 50,000 training images and 10,000 validation images. To implement semi-supervised learning experiments, we chose N_l images randomly as a labeled subset

Table 3: Comparison of error rates (%) between the adversarial regularization (AR) and other published results on the SVHN dataset, with 1000 labels.

	Test error (ave. ($\pm$ std.) %)
Methods	$N_l = 1000$
DGM (2014) [8]	36.02($\pm$0.1)
VAT (2015) [16]	24.63
ADGM (10MC) (2016) [9]	22.86
ALI (2016) [14]	19.14($\pm$0.5)
AAE (2015) [20]	17.70($\pm$0.30)
SDGM (2016) [9]	16.61($\pm$0.24)
Improved-GAN (2016) [13]	8.11($\pm$1.3)
AR (ours)	10.94($\pm$0.27)

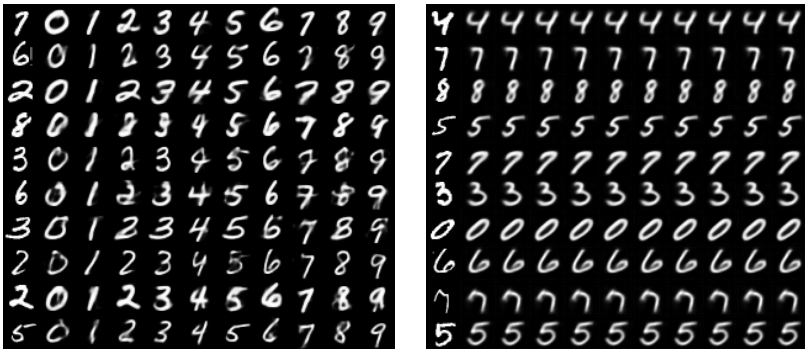

Figure 2: Changing the label $\boldsymbol{y}$ with the clamped style $\boldsymbol{z}$. The left digits are samples from the dataset $p_{data}(\boldsymbol{x})$ for the MNIST database, and the remaining digits are generated with the style $\boldsymbol{z}$ obtained from the leftmost images and the arbitrary label $\boldsymbol{y}$. The left and right images use hyper-parameters for visualization and the best classification, respectively.

$p_{labeled}(\boldsymbol{x})$, where each class has the same number of labeled images. We removed the label information from the remaining $(50,000 - N_l)$. We trained our deep neural networks using the training images and chose hyper-parameters according to the accuracy of the validation images. Then, the final classification error was evaluated on the test images, with the model configured by the chosen hyper-parameters. The SVHN dataset consists of 32-by-32 RGB images of digits in home numbers, comprising 604,388 images for training and 26,032 images for testing. We chose 6,000 validation images randomly from the training images. The remaining conditions were the same as those of the MNIST database. The architectures of the autoencoders were in accordance with preceding studies: [25] for the MNIST database and [11] for the SVHN dataset (see Tables 4 and 5 in Appendix for detail).

We show the classification results of AE [4], AAE [20] and AR on the MNIST database and SVHN dataset (see Tables 1 and 2). The result of AAE is obtained from the original paper [20]. We searched the set of hyper-parameters that achieved the best performance on validation data, and used $\lambda_{cl} = 1$, $\lambda_{rec} = 2000$, and $\lambda_{ar} = 0$ for AE; and $\lambda_{cl} = 1$, $\lambda_{rec} = 1$, and $\lambda_{ar} = 1$ for AR (see Appendix in detail).

We visualized the independence of the latent representations $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$ by changing them [17, 8, 11]. We searched the best hyper-parameters for visualization and used $\lambda_{cl} = 1$, $\lambda_{rec} = 500$, and $\lambda_{ar} = 100$. First, several images were sampled from the dataset $p_{data}(\boldsymbol{x})$ and their styles

$\hat{\boldsymbol{z}}$ were extracted by the encoder q. By using the extracted styles $\hat{\boldsymbol{z}}$ and the arbitrary label $\boldsymbol{y}$, the decoder p generated artificial images (see Fig. 2). The generated images shared their styles, such as penmanship and font type, regardless of the digit types; this suggests that the latent representation $\boldsymbol{y}$ corresponded to digit type independently of style. The right image of Fig. 2 shows the generated images of the hyper-parameters when the model yields the best classification result.

5 Discussion

The autoencoder has been studied as an unsupervised dimension reduction, contributing a penalty to the semi-supervised classification [4, 5, 6, 7]. Studies on VAE introduced a further penalty to the latent representation, based on Kullback-Leibler (KL) divergence, to match the posterior distribution $q(\boldsymbol{z}|\boldsymbol{x})$ of the latent representation $\boldsymbol{z}$ obtained from each input $\boldsymbol{x}$ to the prior distribution $p_{prior}(\boldsymbol{z})$ [8, 9, 17]. However, the KL penalty was calculated image-by-image (see Eq. (2)), which does not guarantee a match between the distribution $q(\boldsymbol{z}|\boldsymbol{x})$ over the latent representations and the prior distribution $p_{prior}(\boldsymbol{z})$. In contrast, attributed to the framework of the GAN, the adversarial regularization provides a stricter penalty: the adversarial regularization gives the distribution $q(\hat{\boldsymbol{y}},\hat{\boldsymbol{z}}|\boldsymbol{x})$ a resemblance to the prior distribution $p_{prior}(\boldsymbol{y},\boldsymbol{z})$ [10, 11, 12, 13, 14]. It guarantees the independence of the label $\boldsymbol{y}$ and the style $\boldsymbol{z}$, and thus obtains the label information $\boldsymbol{y}$ separated from the remaining style information $\boldsymbol{z}$. This is one of the reasons why the adversarial regularization contributes to the semi-supervised classification. Recall that the hyper-parameters for visualization differ from those of classification. We consider that the separation of label and style and the reconstruction of images function well as regularization. However, they are simply penalty terms and are not always objectives compatible with classification.

The discriminator D uses posterior distributions $q(\hat{\boldsymbol{y}},\hat{\boldsymbol{z}}|\boldsymbol{x})$ itself, while the discriminator D draws a sample from the prior $p_{prior}(\boldsymbol{y},\boldsymbol{z})$. Thereby, each element $\hat{y}_k$ of the posterior distribution $q(\hat{\boldsymbol{y}}|\boldsymbol{x})$ of the label $\boldsymbol{y}$ approaches 0 or 1 to mimic the sample from the prior distribution $p_{prior}(\boldsymbol{y},\boldsymbol{z})$, resulting in a decrease in the information $H[q(\hat{\boldsymbol{y}}|\boldsymbol{x})]$. This kind of penalty has been proposed in previous studies [12, 13] and is expected to increase the margin between classes and to promote the unambiguous classification. This is another reason for the success of the adversarial regularization.

The discriminator D used in this study was a tiny three-layered perceptron, regardless of the size of dataset and the architecture of autoencoder, indicating that the computation time for the discriminator D and the adversarial regularization is almost negligible compared with that of the autoencoder consisting of two deep CNNs. Therefore, the total computation time is almost the same as or less than that of the existing semi-supervised learning algorithms.

As shown in Tables 2 and 3, the autoencoder trained with our proposed regularization achieved accuracy comparable with or superior to that of the existing state-of-the-art semi-supervised algorithms for the benchmark semi-supervised tasks. Unfortunately, Improved-GAN surpasses our proposed regularization in both tasks. However, approaches based on GAN, such as ALI and Improved-GAN, are well known to encounter the issue of *mode collapse* [28, 29]. A dataset used for classification problems has a multimodal distribu-tion in the data space because it is a collection of datasets, where each belongs to one of the classes. However, GAN is prone to fail to generate artificial data with a multimodal distribution; more specifically, the generator collapses, generating only a certain mode sample or a

Table 4: Architecture of Autoencoder for the MNIST database.

encoder q	decoder p
28×28 grayscale image	60-D latent representation
conv. (c32, k5, s1), BN, ReLU	fc (n1024), BN, ReLU
max-pool. (s2)	fc (n2048), BN, ReLU
conv. (c64, k5, s1), BN, ReLU	up. (s2)
max-pool. (s2)	deconv. (c64, k5, s1), BN, ReLU
conv. (c128, k5, s1), BN, ReLU	up. (s2)
max-pool. (s2)	deconv. (c32, k5, s1), BN, ReLU
fc (n1024), BN, ReLU, dropout(p0.5)	up. (s2)
fc (n60), BN, ReLU	deconv. (c1, k5, s1), BN, ReLU

small family of very similar samples. While GAN is required to find fine hyper-parameters to model a multimodal distribution, this is difficult in the case of semi-supervised learning because of the limited knowledge of the dataset. Therefore, Improved-GAN's high performance in the semi-supervised classification of MNIST does not guarantee its high performance for unknown datasets used in practical tasks. Conversely, our proposed regularization potentially overcomes this issue because it generates images in the manner of AE (which is free from mode collapse) and only uses GAN to regularize the latent variables, which follow a unimodal distribution (i.e., Gaussian distribution and uniform distribution.) Therefore, we consider our method to be more robust than other GAN-based approaches. A more detailed comparison will be conducted in future. In addition, the discriminator D used in this study was a tiny three-layered perceptron, regardless of the size of dataset and the architecture of the autoencoder. This indicates that the computation time for the discriminator D and our proposed regularization is almost negligible compared with that of the autoencoder consisting of two deep CNNs. Therefore, the total computation time is considerably less than that of the semi-supervised learning algorithms based on deep generative models (e.g., DGM, ADGM, and SDGM), which require a minimum of three deep CNNs.

6 Conclusion

This paper proposed the adversarial regularization of the joint distribution of latent representations of an autoencoder for semi-supervised classification. The adversarial regularization provides a penalty that divides the latent representations into the label information and the remaining style information. Therefore, the autoencoder trained with the regularization achieved an accuracy comparable or superior to the existing state-of-the-art semi-supervised algorithms for the benchmark semi-supervised tasks.

Acknowledgments

This work was partially supported by the JSPS KAKENHI (16K12487) and the SEI Group CSR Foundation.

Table 5: Architecture of Autoencoder for the SVHN dataset.

encoder q	decoder p
28×28 RGB image	60-D latent representation
conv. (c64, k5, s1), BN, lReLU	fc (n512 $\times$ 4 $\times$ 4), BN, ReLU
conv. (c128, k4, s2), BN, lReLU	deconv. (c256, k4, s1), BN, lReLU
conv. (c256, k4, s1), BN, lReLU	deconv. (c128, k4, s2), BN, lReLU
conv. (c512, k4, s2), BN, lReLU	deconv. (c64, k4, s1), BN, lReLU
fc (n60)	deconv. (c3, k4, s2), BN, lReLU

A Details of Experimental Settings

The appendix provides the details of the experimental settings of our results. The architectures of the autoencoder used for the MNIST database and SVHN dataset are shown in Table 4 and Table 5, respectively. "conv." and "deconv." denote convolution and fractionally strided convolution, respectively, outputing c feature maps with a $k \times k$ kernel and stride of s; "max-pool." and "up." denote max-pooling and upscaling, respectively, with a stride of s; "fc" denotes matrix multiplication outputting an n-dimensional vector; "BN" denotes batch normalization; "ReLU" and "sigmoid" are activation functions; and "lReLU" is the leaky ReLU activation function with a slope of 0.01. The autoencoder and discriminator were trained using the Adam optimization algorithm [27] with a weight decay of 0.0005, where the parameter α was set to $\alpha = 0.0001$ for the experiments detailed in Section 4.1 The MNIST database and SVHN dataset had batch sizes of 500 and 100, respectively.

The coefficients λ_{rec} and λ_{ar} of the errors $\mathscr{L}_{rec}$ and $\mathscr{L}_{ar}$, respectively, were grid-seared over the range of $\{\ldots, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0, \ldots\}$, where the coefficient λ_{cls} was set to 1.

References

[1] J. Lasserre *et al.*, IEEE, vol. 1, no. 6., pp. 87–94, 2006.

[2] Y. LeCun, *et al.*, *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] J. Schmidhuber, *Neural Networks*, vol. 61, pp. 85–117, 2015.

[4] G. E. Hinton and R. R. Salakhutdinov, *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[5] M. Ranzato and M. Szummer, *ICML*, pp. 792–799, 2008.

[6] X. Zhao and P. a. Robinson, *Journal of Computational Neuroscience*, pp. 197–216, 2015.

[7] A. Rasmus *et al.*, *NIPS*, pp. 3546–3554, 2015.

[8] D. P. Kingma, *et al.*, *NIPS*, pp. 3581–3589, 2014.

[9] L. Maaløe *et al.*, *ICML*, vol. 48, pp. 1–5, 2015.

[10] I. J. Goodfellow *et al.*, *NIPS*, pp. 2672–2680, 2014.

[11] A. Radford, *et al.*, *ICLR*, pp. 1–16, 2016.

[12] J. T. Springenberg, *arXiv:1511.06390*, 2015.

[13] T. Salimans *et al.*, *arXiv:1606.03498*, 2016.

[14] V. Dumoulin *et al.*, *arXiv:1606.00704*, 2016.

[15] S. Rifai *et al.*, *NIPS*, pp. 2294–2302, 2011.

[16] T. Miyato *et al.*, *ICLR*, pp. 1–18, 2016.

[17] D. P. Kingma and M. Welling, *ICLR*, pp. 1–14, 2014.

[18] S. Nowozin, *et al.*, *arXiv:1606.00709*, 2016.

[19] M. Arjovsky and L. Bottou, *NIPS*, pp. 1–16, 2017.

[20] A. Makhzani *et al.*, *arXiv:1511.05644*, 2015.

[21] R. Tachibana, *et al.*, *ICIS*, pp. 939–944, 2016

[22] A. B. L. Larsen, *et al.*, *ICML*, 2016.

[23] "THE MNIST DATABASE of handwritten digits." url: `http://yann.lecun.com/exdb/mnist/`

[24] "The Street View House Numbers (SVHN) Dataset." url: `http://ufldl.stanford.edu/housenumbers/`

[25] "Deep MNIST for Experts," in *TensorFlow Turorials*. url: `https://www.tensorflow.org/tutorials/mnist/pros/`

[26] B. Cheung *et al.*, *ICLR*, 2015.

[27] D. Kingma and J. Ba, *ICLR*, 2015.

[28] X. Mao *et al.*, *arXiv:1611.04076*, 2017.

[29] L.Metz *et al.*, *ICLR*, 2017.