MCMC based Generative Adversarial Networks for Handwritten Numeral Augmentation

He Zhang[1], Chunbo Luo[1], Xingrui Yu[2], and Peng Ren[2]

[1] College of Engineering, Mathematics and Physical Sciences, University of Exeter,
EX4 4QF, Exeter, United Kingdom

[2] College of Information and Control Engineering, China University of Petroleum
(East China), 266580, Qingdao, China

Abstract. In this paper, we propose a novel data augmentation framework for handwritten numerals by incorporating the probabilistic learning and the generative adversarial learning. First, we simply transform numeral images from spatial space into vector space. The Gaussian based Markov probabilistic model is then developed for simulating synthetic numeral vectors given limited handwritten samples. Next, the simulated data are used to pre-train the generative adversarial networks (GANs), which initializes their parameters to fit the general distribution of numeral features. Finally, we adopt the real handwritten numerals to fine-tune the GANs, which increases the authenticity of generated numeral samples. In this case, the outputs of the GANs can be employed to augment original numeral datasets for training the follow-up inference models. Considering that all simulation and augmentation are operated in 1-D vector space, the proposed augmentation framework is more computationally efficient than those based on 2-D images. Extensive experimental results demonstrate that our proposed augmentation framework achieves improved recognition accuracy.

Keywords: Data augmentation, probabilistic model, generative adversarial learning, handwritten numeral classification.

1 Introduction

Handwritten numeral recognition is one of the most extensively studied topics in pattern recognition and machine learning. It has been applied to various tasks include tax forms processing [1], postal mail sorting and bank check reading [2], etc. However, real world handwritten numerals exhibit significant difference due to the writing styles and habits of different human beings. In this case, inferring handwritten numerals in emerging styles becomes a severe challenge to conventional numeral recognition system.

Data augmentation, which could effectively enlarge the training set based on the limited training samples, has been proposed as one highly potential solution to tackle the challenge mentioned above. In general, existing data augmentation approaches for enriching imbalanced training sets can be categorized into two types: sampling based augmentation and cropping based augmentation. *Sampling based approaches*, such as Markov chain Monte Carlo (MCMC) [3] and its variant Gibbs sampling [4], rely on simulating samples from the reconstructed probabilistic distribution of the target data set. Unfortunately, features of emerging style of handwritten numerals cannot be precisely modeled by probabilistic distribution. In addition, the simulated numeral data lacks of authenticity to represent real ones. *Cropping based approaches* operate a series of image transformations (e.g., rotation and cropping) to generate extra images or image patches. Whereas, this approach do not augment training image set in a substantial manner.

Recently, game theory inspired generative adversarial networks (GANs) [5] have been proposed as unsupervised deep learning methods with the exceptional data representation power in computer vision [6]. The *Discriminator* and the *Generator* (i.e., two independent players) of the GANs take in real data samples as well as variables of random noise for adversarial training, respectively. The *Generator* then outputs data samples that are closely related to the real data. In this case, the GANs can be viewed as data-driven generative models for data augmentation. Unfortunately, current adversarial models exhibit a range of limitations such as training uncertainty and mode collapse (i.e., artifacts and unrealistic outputs), which prevent them for real world data augmentation applications. Furthermore, conducting convolutional operations on 2-D digital images brings about heavy computational loads of training model parameters.

In this paper, we propose a probabilistic sampling based GANs for data augmentation, which tackles the aforementioned challenges of augmenting emerging handwritten numerals. Providing that we have transformed a limited collection of new style handwritten numerals into vector form for training a classification model (e.g., support vector machine or logistic regression). The proposed algorithm tries to increase the number of the emerging handwritten numerals of each category by sampling from Gaussian based MCMC probabilistic model. We then explore the benefits of competition based data-driven GANs to increases the authenticity of the simulated numeral data based on 1-D real handwritten numeral features. In this case, the MCMC based approach provides initial inputs for pre-training the GANs to fit their parameters on general handwritten numeral distribution. Incorporating real emerging numerals, the model parameters of GANs are then adaptively fine-tuned. After the model reaches the convergence state, the GANs can produce high quality augmented data for the follow-up classification systems. Comprehensive experiments confirm the effectiveness of the proposed framework for augmenting handwritten numerals.

2 Review of MCMC and GANs

In this section, we firstly introduce the principal of MCMC based sampling methods for data simulation. We then describe the main formulation of competition inspired generative adversarial networks.

2.1 MCMC based Sampling Methods

In statistical learning, MCMC [3] has played a significant role in approximating a hard combinatorial problem by constructing a Markov chain that has the target combinatorial distribution as the invariant distribution [4]. In this subsection, we briefly introduce two main MCMC based sampling methods which are widely employed in academia and industry.

Metropolis-Hastings (MH) algorithm The MH algorithm [7] is the general foundation of many advanced MCMC algorithms (e.g., the simulated annealing and the Gibbs sampler [8]). The MH algorithm tries to simulate a candidate given the current sample according to an easy-to-sample proposal distribution. Meanwhile, the Markov chain moves to candidate with the acceptance probability. Otherwise, it remains at current one. One critical limitation of the MH and its related algorithms is that the convergence speed of sampling is not guaranteed since the Markov chain is problem-dependent [4].

Gibbs Sampling To address the convergence problems in MH inspired MCMC algorithms, Geman and Geman [8] proposed Gibbs sampling, in which a full conditional distributions are adopted for designing proposal distribution [4]. However, the full conditionals might not available for real world data such as handwritten numerals. Considering the trade-off between efficiency and accuracy, we employ the Gaussian distribution to model handwritten numeral samples and adopt the Gibbs sampler for data simulation.

2.2 Generative Adversarial Networks

The game inspired GANs for data representation learning was first proposed by Goodfellow et al. [5] and immediately applied to computer vision [6] [9] and NLP [10]. The initial GANs can be treated as a non-cooperative two-player game: one player is referred to the *Discriminator* and another is considered as the *Generator*. During the evolution of the game, the *Generator* tries to learn from the real data distribution and produce forged data with random inputs. On the contrary, the *Discriminator* takes in both the forged data and the real ones, and learns to distinguish the counterfeit from the truth [5].

Given that we have a collection of real data x and another collection of random variables z, we denote $p_{\text{real}}(x)$ and $p_{\text{noise}}(z)$ as their corresponding probabilistic distributions. The adversarial learning between the *Discriminator* and the *Generator* can be formulated as the following optimization problem:

$$\begin{aligned} &\min\nolimits_G \max\nolimits_D \ V(D,G) \\ &= \mathbb{E}_{x \sim p_{\text{real}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_{\text{noise}}(z)}[\log(1 - D(G(z)))], \end{aligned} \tag{1}$$

where D and G represent the *Discriminator* and the *Generator*, respectively. In this case, those two models fight against each other during training stage which results in a potent generative model G.

However, the generative model formulated above is reported to remain remarkable difficulties in training [11] [12]. First, the stability and certainty of training GANs are sensitive to modifications. In other words, the updates get worse for the *Generator* while the *Discriminator* get better due to the significant difference of convergence speed among them. On the contrary, if the *Generator* learns faster than the *Discriminator*, the GANs might produce unrealistic samples compared with the existing real data. Second, the GANs usually require a large amount of real samples for training. In our handwritten numeral generation case, training GANs with the limited emerging numeral data is easy to cause overfit problem, which means the augmented numerals are meaningless to train the following supervised inference model.

3 Probabilistic Model based Data Simulation

In this section, we employ the Gaussian based probabilistic model for estimating the real distributions of handwritten numerals in vector space, which is then adopted as the proposal distribution to simulate numeral samples by using Gibbs sampler.

Let $\boldsymbol{X} \in \mathbb{R}^{M \times c}$ be the matrix of one class of emerging handwritten numerals, where M is the number of data observations and c represents the feature dimensions. Then, each sample of $\boldsymbol{X}$ denoted as x can be modeled as a Gaussian variable with the density function and its exponential form below:

$$\begin{aligned} &\text{Gaussian}(x;\mu,\sigma) \\ &= \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\ &= \exp(-\text{log}(\sigma\sqrt{2\pi}) - \frac{(x-\mu)^2}{2\sigma^2}), \end{aligned} \tag{2}$$

where μ and σ are the mean and variance in Gaussian distribution. Considering that the conjugate distribution of Gaussian distribution is itself, we thus model μ and σ as the mean and variance of the given emerging handwritten numerals plus a standard normal distribution $\mathcal{N}(0,1)$, respectively.

Based on Bayes theory [13], we have the following relationship between posterior conditionals $p(\mu,\sigma|\boldsymbol{X})$ and its full joint distribution:

$$\begin{aligned} p(\mu,\sigma|\boldsymbol{X}) &\propto p(\boldsymbol{X}|\mu,\sigma)p(\mu)p(\sigma) \\ &\propto \text{log}(\prod_{i=1}^{M} p(x^{(i)}|\mu,\sigma)) + \text{log}p(\mu) + \text{log}p(\sigma), \end{aligned} \tag{3}$$

where each of the log term is actually the density function of the Gaussian distribution. In this case, we can employ the designed Gaussian distribution to sample the parameter μ and σ and further simulate the numeral samples from the Gaussian distribution using Gibbs sampling.

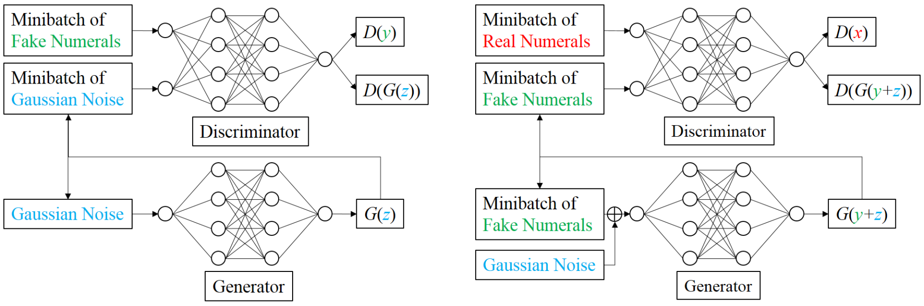
Minibatch of Fake Numerals
Minibatch of Gaussian Noise
D(y)
D(G(z))
Discriminator
Gaussian Noise
G(z)
Generator
Minibatch of Real Numerals
Minibatch of Fake Numerals
D(x)
D(G(y+z))
Discriminator
Minibatch of Fake Numerals
Gaussian Noise
G(y+z)
Generator

Fig. 1. The GANs for handwritten numeral augmentation.

4 GANs based Data Augmentation

In this section, we begin by introducing the formulation of our competition based GANs for numeral augmentation. Then, we provide detailed training and optimizing schemes of the proposed framework. To make it easier to be understood, we adopt identical notations as previous sections in our formulation.

4.1 Generative Adversarial Model Formulation

Let $p_{\text{real}}(x)$ and $p_{\text{fake}}(y)$ be the probabilistic distributions of handwritten numeral data x and simulated data y, respectively. We denote D and G as two multi-layer perceptions (MLPs). The goal of our proposed model is to augment the simulated data collection with increased authenticity by two-fold adversarial training between D and G. The *pre-training* stage is formulated as below:

$$\begin{aligned}&\min\nolimits_G \max\nolimits_D \ V(D,G)\\ =\ &\mathbb{E}_{y\sim p_{\text{fake}}(y)}[\log D(y)] + \mathbb{E}_{z\sim p(z)}[\log(1-D(G(z)))],\end{aligned} \tag{4}$$

where z is the Gaussian white noise. In this case, a large amount of simulated data are used to pre-train the GANs, which initializes the model parameters to fit the general representation of handwritten numerals. After the convergence of the pre-training stage, we then operate the *fine-tuning* stage with limited handwritten numerals as the following formulation:

$$\begin{aligned}&\min\nolimits_G \max\nolimits_D \ V(D,G)\\ =\ &\mathbb{E}_{x\sim p_{\text{real}}(x)}[\log D(x)] + \mathbb{E}_{y\sim p_{\text{fake}}(y)}[\log(1-D(G(y+z)))].\end{aligned} \tag{5}$$

During the fine-tuning process, we plus the Gaussian white noise on simulated data as the input for G. The reasons that we choose $G(y+z)$ rather than $G(y)$ or $G(z)$ are manifold. First, incorporating simulated numeral features with random variables compensates the generalization ability of training adversarial model using limited real handwritten samples. Second, inducing randomness could reduce the possibility of overfitting, especially training complex inference models like MLPs with limited handwritten numerals. Figure 1 illustrates the two-fold training scheme of our method.

4.2 Training and Optimization Strategies

Let $D(\theta_d)$ and $G(\theta_g)$ denote the *Discriminator* and the *Generator*, with θ_d and θ_g be the corresponding hyperparameters of the MLPs, respectively. The goal of training $D(\theta_d)$ and $G(\theta_g)$ equals to optimize the objective functions in Eq. (4) and Eq. (5). Since D and G are two artificial neural networks (ANNs), we then adopt the minibatch based stochastic gradient algorithm to optimize the hyperparameters θ_d and θ_g.

According the objective functions described in Eq. (4) and Eq. (5), the gradient based approach of optimizing the hyperparameter of D and G requires computing the stochastic gradients of $V(D,G)$ with respect to θ_d and θ_g. For a minibatch of m data samples and $m \leq \min\{M,N\}$, the minibatch stochastic gradients of D in pre-training and fine-tuning stage can be computed as follows:

$$\frac{\partial V}{\partial \theta_d} = \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(\cdot^{(i)}) + \log(1 - D(G(\cdot^{(i)})))], \tag{6}$$

and the minibatch stochastic gradients of G in pre-training and fine-tuning stage can be similarly computed as below:

$$\frac{\partial V}{\partial \theta_g} = \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [\log(1 - D(G(\cdot^{(i)})))]. \tag{7}$$

After obtaining the gradients $\partial V/\partial \theta_d$ and $\partial V/\partial \theta_g$, advanced gradient based approaches can be employed to efficiently optimize hyperparameters between D and G on CPU machine, since all data are formulated in vector space. We use ADAM [14] to train our GANs in all experiments.

5 Experimental Validation

In this section, we provide detailed experimental validations of the proposed handwritten numeral augmentation framework. All experiments are performed on a PC machine with Intel i7-5930K CPU (3.50GHz) and 32 GB memory.

5.1 Handwritten Numeral Data

The MNIST and the USPS data sets are used in our experiments. Based on the assumption that the training data of emerging style of handwritten numerals is limited, we randomly select two training subsets from the original data sets with each contains 300 numeral samples per class. Those two subsets are both used to train the SVM and LR models and data simulation and generation frameworks.

Table 1. Classification Accuracy of Inference Models

	Subset	Sub.+MCMC	Sub.+GANs
MNIST (SVM)	91.74	91.76	92.50
MNIST (LR)	89.95	89.91	91.20
USPS (SVM)	92.58	92.53	93.68
UPSP (LR)	91.03	91.38	91.74

5.2 Parameter Setting

The mini-batch size is set as 50 to balance the trade-off between computation expenses and over-fitting risks. The two MLPs in GANs are equally designed as 3 layers with 30 hidden nodes in each hidden layer. The learning rates of the *Discriminator* and the *Generator* are both set as 2×10^{-4}. Furthermore, we use momentum [15] to accelerate the training speed and set its parameters as 0.9 and 0.99 in both pre-training and fine-tuning stages.

5.3 Result Analysis

We show the testing accuracy of different classification models in Table 1. The comparable classification accuracy of the SVM and LR models trained on the data set augmented by GANs strongly confirms the effectiveness of our proposed handwritten numeral augmentation framework.

6 Conclusion

In this paper, a novel probabilistic and adversarial learning based data augmentation framework has been proposed for augmenting emerging handwritten numerals in supervised recognition system. Our algorithm combines the Gaussian probabilistic model to approximate the distributions of emerging handwritten numerals and the MCMC based sampling for simulation. The two-fold data-driven GANs are then trained to enhance the authenticity of augmented numerals in 1-D vector space. Experimental validations prove the effectiveness of the proposed framework for handwritten numeral augmentation.

Acknowledgments. This work was supported by grants from the Chinese Scholarship Council (CSC) program and partially sponsored by the Newton Fund Researcher Links Programme Workshop and University of Exeter Outward Mobility Academic Fellowship.

References

1. Ha, T.M., Bunke, H.: Off-Line, Handwritten Numeral Recognition by Perturbation Method. IEEE Trans. Pattern Anal. Mach. Intell. 19, 535-539 (1997)

2. Wu, Y.C., Yin, F., Liu, C.L.: Evaluation of Geometric Context Models for Handwritten Numeral String Recognition. In: 14th ICFHR, pp. 193-198. IEEE Press, Greece (2014)
3. Gelfand, A.E., Smith, A.F.: Sampling-based Approaches to Calculating Marginal Densities. J. Am. Stat. Assoc. 85, 398-409 (1990)
4. Andrieu, C., de Freitas, N., Doucet, A., Jordan, M.I.: An Introduction to MCMC for Machine Learning. Mach. Learn. 50, 5-43 (2003)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: 27th NIPS, pp. 2672-2680. Curran Associates, Inc., Canada (2014)
6. Denton, E.L., Chintala, S., Fergus, R.: Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In: 28th NIPS, pp. 1486-1494. Curran Associates, Inc., Canada (2015)
7. Metropolis, N., Ulam, S.: The monte carlo method. J. Am. Stat. Assoc. 44, 335-341 (1949)
8. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Trans. Pattern Anal. Mach. Intell. 6, 721-741 (1984)
9. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A. A.: Context Encoders: Feature learning by inpainting. In: 2016 CVPR, pp. 2536-2544. IEEE Press, America (2016)
10. Dai, B., Lin, D., Urtasun, R., Fidler, S.: Towards Diverse and Natural Image Descriptions via a Conditional GAN. In: arXiv preprint arXiv:1703.06029 (2017)
11. Arjovsky, M., Bottou, L.: Towards Principled Methods for Training Generative Adversarial Networks. In: arXiv preprint arXiv:1701.04862 (2017)
12. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. In: arXiv preprint arXiv:1703.06029 (2017)
13. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification. John Wiley and Sons (2012)
14. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. In: arXiv preprint arXiv:1412.6980 (2014)
15. Sutskever, I., Martens, J., Dahl, G. E., Hinton, G. E.: On the importance of initialization and momentum in deep learning. In: 30th ICML, pp. 1139-1147. PMLR, America (2013)