Postprint of Image Classification Algorithm Based on Feature-Recalibration Generative Adversarial Networks

Authors: Jiang Daihong, Zhang Sanyou, Liu Qikai

Date: 2019-01-03T00:00:00+00:00

Abstract

To address the limitation that traditional discriminator loss strategies and architectures struggle to extract more abstract and task-relevant robust features, resulting in inadequate performance for semi-supervised image classification, we propose a feature recalibration-based generative adversarial network. To learn task-relevant features, we introduce, building upon existing semi-supervised GANs, an unsupervised mean squared error loss regularization term for the discriminator across different model states, which imposes parameter penalties on different outputs corresponding to the same input from two branches in the training samples, thereby guiding the optimization direction of feature recalibration. Furthermore, we incorporate a squeeze-and-excitation module into the discriminator to optimize the conventional convolutional pooling architecture. This module automatically learns the importance of each feature channel, enabling the extraction of task-relevant features while suppressing task-irrelevant features, and implements the feature recalibration function, consequently enhancing the performance of semi-supervised image classification.

Full Text

Preamble

Vol. 37 No. 3
Application Research of Computers
Accepted Paper

Image Classification Algorithm Based on Feature Recalibration Generative Adversarial Networks

Jiang Daihong[1], Zhang Sanyou[2,3], Liu Qikai[3]
(1. School of Information & Electronic Engineering, Xuzhou Institute of Tech-

nology, Xuzhou Jiangsu 221008, China;
2. Dept. of Science & Technology, Suzhou Wujiang District Public Security Bureau, Suzhou Jiangsu 215200, China;
3. School of Information & Control Engineering, China University of Mining & Technology, Xuzhou Jiangsu 221008, China)

Abstract: Traditional discriminator loss strategies and architectures struggle to extract more abstract and task-related robust features, leading to inadequate performance in semi-supervised image classification. To address this, we propose a feature recalibration generative adversarial network. To learn task-related features, we introduce an unsupervised mean squared error loss regularization term for the discriminator under different model states, based on existing semi-supervised GANs. This regularization penalizes different outputs corresponding to the same input from two branches of training samples, thereby guiding the optimization direction of feature recalibration. Additionally, we incorporate a squeeze-and-excitation module into the discriminator to optimize the conventional convolutional pooling structure. This module automatically learns the importance of each feature channel, enabling extraction of task-related features while suppressing task-irrelevant ones, thereby achieving feature recalibration and improving semi-supervised image classification performance.

Keywords: generative adversarial network; image classification; feature recalibration; deep learning

0 Introduction

Image classification is one of the most fundamental tasks in computer vision, and semi-supervised image classification algorithms can effectively address performance limitations when labeled samples are scarce. In recent years, convolutional neural networks (CNNs) have elevated visual tasks to new heights through image feature extraction. Kingma et al. [**?**] proposed a flexible and scalable deep generative network algorithm that employs rich parametric density estimation for modeling and variational inference for training, including the implicit feature discriminative model (M1), generative semi-supervised model (M2), and a stacked semi-supervised model (M1+M2), which achieved favorable classification performance. Maaløe et al. [**?**] extended the deep generative network model by proposing auxiliary deep generative models (ADGM), which improve variational approximation algorithms by introducing auxiliary latent variables, making the variational approximation more expressive. Li et al. [**?**] proposed a max-margin deep generative model (MMDGM) that enhances the predictive performance of generative models while retaining strong generative capabilities by treating the predictions of a margin classifier as missing labels. Rasmus et al. [**?**] proposed the ladder network based on autoencoders, where the encoding network performs supervised learning and each layer of the decoding network corresponds to the encoder for unsupervised training, forming a ladder struc-

ture that uses reconstruction loss at each layer as the total loss function to guide parameter optimization. Laine et al. [**?**] proposed simple and efficient deep neural models (Π-model and Temporal ensembling model) that borrow self-learning concepts, using predictions on unlabeled data samples to guide model parameter optimization. Specifically, they introduced unsupervised regularization terms for supervised loss under different data augmentation conditions, achieving good results in semi-supervised image classification.

GAN neural network models demonstrate promising performance in semi-supervised learning. Miyato et al. [**?**] proposed a regularization method based on virtual adversarial training (VAT): local distribution smoothing (LDS). Salimans et al. [**?**] improved conventional GAN performance through several techniques, extending the discriminator' s binary probability output to multi-class probability output, where the total categories consist of real sample classes plus the generated sample class, thereby enabling prediction. Springenberg et al. [**?**] proposed Cat-GAN, which introduces an appropriate objective function in the discriminator to balance the mutual information between input samples and their predicted categories. The classifier is trained by minimizing the conditional cross-entropy of real data sample categories and maximizing that of generated data categories, thereby improving classifier performance. Li et al. [**?**] proposed triple GAN, which introduces an additional classifier to address the issue where the generator and discriminator in GAN cannot simultaneously achieve optimal performance in semi-supervised learning, and enables the generator to learn semantic features of samples. Dumoulin et al. [**?**] proposed the adversarially learned inference (ALI) model, which introduces an inference network that maps training samples from data space to latent variable space, outputting the joint distribution of training samples and latent variables, while the generator outputs the joint distribution of latent variables and generated samples.

Model depth and spatial dimension are two important aspects for improving CNN performance. Jaderberg et al. [**?**] introduced the attention mechanism into the spatial dimension, allowing spatial manipulation of data within the network and enabling neural networks to actively perform feature mapping in space. In addition to improving network performance in spatial dimensions, feature channels represent another important breakthrough. Hu et al. [**?**] proposed the squeeze-and-excitation network (SENet) module, which can explicitly model interdependencies between feature channels to extract useful features while suppressing task-irrelevant features.

Therefore, to improve model generalization capability and semi-supervised classification performance, we propose a feature recalibration generative adversarial network that introduces unsupervised mean squared error loss for the discriminator under different model states, penalizing different outputs obtained from the same input. Simultaneously, we incorporate the SENet module into the discriminator to automatically learn the importance of each feature channel and perform feature recalibration, thereby enhancing semi-supervised learning

performance.

1 Feature Recalibration Generative Adversarial Network Structure

The structure of the feature recalibration generative adversarial network is shown in Figure 1 [Figure 1: see original paper]. The generator takes random noise of dimension 100 as input, first mapping it to a high-dimensional space of 1024×16, then reshaping to channel dimensions, and performing deconvolutional network (DN) operations. Each deconvolution operation doubles the feature map size, ultimately producing generated samples with the same dimensions as the input images. DN is an unsupervised algorithm for mid- and low-level feature learning proposed by Zeiler et al. [**?**], which constructs mid- and low-level feature maps of input images by learning parameters of sparse coding. Essentially, it learns the mapping matrix between the input image' s latent space feature maps and corresponding convolution kernels, then reconstructs the input image using the learned kernels.

The discriminator receives both generated samples and real samples, with real samples consisting primarily of unlabeled samples and a limited number of labeled samples. Labeled samples only participate in the discriminator' s supervised loss, while unlabeled and generated samples participate in both unsupervised adversarial loss and unsupervised mean squared error loss. Input samples first undergo several convolution and pooling operations to obtain a feature map, followed by a SENet module for feature recalibration to obtain recalibrated feature maps. To increase network nonlinear fitting capability, we introduce the NIN network, which essentially performs 1×1 convolutions. To reduce model parameters and prevent overfitting, we use global average pooling instead of fully connected layers, ultimately weighting to obtain the total loss.

2 Image Classification Algorithm Based on Feature Recalibration GAN

This paper constructs a feature recalibration generative adversarial network from the perspectives of model ensemble principles and model structure. During training, due to the presence of dropout regularization and random noise in the network model, even with the same input sample at the same training iteration, network outputs may vary. Therefore, within the same batch iteration, random processing of the same data will also change its output. To guide the direction of model parameter optimization, we desire that the same input produces outputs that are as similar as possible, meaning the probabilities of belonging to a particular class should be as close as possible for classification predictions.

2.1 Algorithm Principle

Let $z \in \mathbb{R}^{100}$ denote a random noise vector of dimension 100. Let $G(z)$ represent generated samples, and let the discriminator' s output represent the probability that its input sample belongs to real samples, denoted as $D(x)$. Let $X = \{x_1, x_2, \ldots, x_m\}$ represent a set of m real samples. Then the overall objective function of conventional GAN can be expressed as:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]$$

Semi-supervised learning in GAN is primarily achieved through the discriminator, which must consider not only the probability that input samples belong to real samples but also the probability of labeled input samples belonging to their label categories. Assuming there are K given label categories, when sample x undergoes feature extraction to obtain a K-dimensional feature vector, the probability of this sample belonging to class j is represented as $p(y = j|x)$. In supervised learning algorithms, the objective function minimizes the cross-entropy between sample labels and model predictions. To obtain GAN' s supervised loss, we define the category of generated samples input to the discriminator as class $K+1$, with the probability from generated samples represented as $p(y = K+1|x)$. Assuming half of the discriminator' s training samples come from real samples and half from generated samples, the discriminator D loss function for semi-supervised GAN can be expressed as:

$$L = L_{\text{supervised}} + L_{\text{unsupervised}}$$

where the supervised loss contributed by labeled training samples corresponds to $L_{\text{supervised}}$, and the unsupervised adversarial loss contributed by unlabeled training samples corresponds to $L_{\text{unsupervised}}$.

For semi-supervised classification tasks, FR-GAN introduces unsupervised feature-level mean squared error loss for the discriminator under different model states, penalizing different outputs obtained from the same input across two branches of real samples. This regularization term can learn task-related features and improve semi-supervised classification accuracy. Let D_{θ_1} and D_{θ_2} represent different states of the discriminator within the same training stage, with $x_i^{(1)}$ denoting unlabeled training samples from the first branch and $x_i^{(2)}$ denoting unlabeled training samples from the second branch. Through random preprocessing function f, the output corresponding to input $x_i^{(2)}$ is $D_{\theta_2}(f(x_i^{(2)}))$, and similarly, the output corresponding to input under the first branch is $D_{\theta_1}(f(x_i^{(1)}))$. The unsupervised feature mean squared error regularization term is expressed as follows:

$$L_{\text{consistency}} = \frac{w(t)}{m} \sum_{i=1}^{m} \| D_{\theta_1}(f(x_i^{(1)})) - D_{\theta_2}(f(x_i^{(2)})) \|^2$$

where $w(t)$ represents the dynamic contribution of the regularization term to the discriminator during training. In the early training stages, due to poor network parameters, the contribution of this regularization term is 0; as training progresses, its contribution gradually increases and finally stabilizes.

Next, we discuss how the SENet module extracts task-related features and how it is embedded into the network for training. Figure 2 [Figure 2: see original paper] shows the structure of the SENet module.

Given an input $X \in \mathbb{R}^{W\times H\times C}$, through a series of convolutional mappings, we obtain a set of C feature maps $U = \{u_1, u_2, \dots, u_C\}$, where $u_c \in \mathbb{R}^{W\times H}$. We then recalibrate the mapped features through three operations:

Squeeze Operation: To explore dependencies between feature channels, we first consider learning information from each channel of the feature maps. Since convolutional kernels operate within a local receptive field, nodes after convolutional transformation cannot obtain information outside this local receptive field. The squeeze operation compresses global spatial information into a channel descriptor by generating statistics through global average pooling. The c-th element of the statistics is calculated as:

$$z_c = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} u_c(i, j)$$

where z_c represents information from the entire image.

Excitation Operation: This is a gating mechanism similar to recurrent neural networks, consisting of two fully connected layers. It first reduces the feature dimension to 1/16 of the input, then after ReLU activation, restores it to the original dimension through another fully connected layer, and finally obtains normalized weights between 0 and 1 through a Sigmoid gate. The specific expression is:

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2\delta(W_1 z))$$

where $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ and $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$, and parameter W explicitly models correlations between feature channels.

Scale Operation: The weights obtained from the excitation operation serve as importance indicators for feature channels after feature selection. These weights are then multiplied element-wise with previous channels to complete recalibration of original features along the channel dimension. The weighting formula for the c-th feature channel is:

$$\tilde{x}_c = F_{scale}(u_c, s_c) = s_c \cdot u_c$$

2.2 Algorithm Flow

The semi-supervised classification algorithm flow based on feature recalibration GAN is shown in Algorithm 1. Let f denote the input data random preprocessing function; L denote the labeled dataset; B denote the dataset composed of each batch size; $w(t)$ denote the dynamic time function for unsupervised weights; and parameters are initialized using the Xavier method.

Initialization: Batch size $m = 100$, representing the number of samples required for each parameter update; iterations: the number of batch iterations, calculated as the total number of samples divided by m; epoch: one epoch equals one complete traversal of the training set.

Hyperparameters: $k = 1$, meaning the discriminator is trained k times before training the generator once; noise samples are 100-dimensional vectors following $\mathcal{U}[-1, 1]$.

The algorithm proceeds as follows:

For each training iteration: - For k steps: - Sample m noise samples $\{z^{(1)}, z^{(2)}, ..., z^{(m)}\}$ - Sample m data samples $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$ - Compute discriminator adversarial loss: $L_{\text{adv}} = \frac{1}{m}\sum_{i=1}^{m}[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]$ - Compute discriminator outputs for real samples under different random states: $B \leftarrow D_{\theta_1}(f(x_i))$ and $B \leftarrow D_{\theta_2}(f(x_i))$ - Compute discriminator supervised and unsupervised losses - Compute discriminator gradient: $\nabla_{\theta_d} = \partial(L_{\text{adv}} + L_{\text{label}})/\partial\theta_d$ - Update discriminator parameters: $\theta_d \leftarrow \text{Adam}(\text{grad}, \theta_d)$ - Sample m noise samples $\{z^{(1)}, z^{(2)}, ..., z^{(m)}\}$ - Compute generator log loss: $L_{\text{gen}} = -\log(D(G(z)))$ - Compute gradient and update generator parameters: $\nabla_{\theta_g} = \partial L_{\text{gen}}/\partial\theta_g$, $\theta_g \leftarrow \text{Adam}(\text{grad}, \theta_g)$

---

3 Experimental Results and Analysis

This chapter validates the proposed method through experiments on the SVHN dataset [**?**]. Experiments were implemented on the Theano [**?**] deep learning framework and run on a single GTX1080 GPU. The generator in our experiments is based on the DCGAN generator, while the discriminator incorporates the SENet module to increase model nonlinearity. The initial learning rate is set to 0.0003, and the model also employs weight normalization (WN), batch normalization (BN), and dropout strategies. Network model parameter configurations are shown in Table 1 .

Table 1: Network Parameter Configuration for SVHN Dataset

Discriminator D	Generator G
Input: 32×32 3-channel	Noise: $\mathbb{R}^{100}$
3×3 conv, 64, lReLU, WN	MLP 8192 units, ReLU, BN

Discriminator D	Generator G
3×3 conv, 64, lReLU, WN	Reshape 512×4×4
3×3 conv, 64, lReLU, WN	5×5 deconv, 256, stride 2, ReLU, BN
Dropout=0.2	5×5 deconv, 128, stride 2, ReLU, BN
3×3 conv, 128, lReLU, WN	5×5 deconv, 3, stride 2, Tanh, WN
3×3 conv, 128, lReLU, WN	
3×3 conv, 128, lReLU, WN	
Dropout=0.2	
SENet block	
3×3 conv, 256, lReLU, WN	
NIN, 128, lReLU, WN	
Global pool layer	
Dense layer 10 Units with WN	

To compare generated sample quality and semi-supervised classification accuracy before and after adding the module, all other model configurations remain identical. For generated sample quality, FR-GAN is compared with a GAN that has only the regularization term without the SENet module. Visual comparison results are shown in Figure 3 [Figure 3: see original paper]. Quantitative comparison using the IS [**?**] (Inception Score) metric is presented in Table 2 .

Table 2: Inception Scores of Samples Generated by Different Models

Method	Real Samples	Score
GAN without SENet		2.89 ± 0.02
GAN without regularization		3.02 ± 0.14
FR-GAN		3.24 ± 0.03

For semi-supervised learning, we set the number of labeled samples per class to 50 and 100, corresponding to 500 and 1000 labeled training samples respectively. As shown in Table 3 , compared with traditional semi-supervised learning algorithms, FR-GAN performs better than most algorithms except for the method in [**?**]. While [**?**] focuses solely on classification tasks and achieves better classification performance, FR-GAN' s loss strategy enables it to outperform conventional GANs in image generation tasks despite being slightly inferior in pure classification.

When training with all labeled samples, assigning incorrect labels to some correctly labeled samples demonstrates that the proposed model can reduce the degree of classification accuracy degradation, preventing significant performance drops and thereby improving network fault tolerance.

As shown in Figure 4 [Figure 4: see original paper], (a) illustrates classification accuracy variation for conventional GAN under certain proportions of incorrect

labels, where accuracy drops rapidly as the proportion increases; (b) shows FR-GAN' s accuracy variation under the same conditions, where accuracy remains essentially stable even when incorrect labels account for 50% of the data.

Table 3: Test Error Rate Comparison on SVHN Dataset

Method	Test Error Rate (%)	#Labels
M1+M2 [**?**]	36.02 ± 0.10	
VAT [**?**]	16.61 ± 0.24	
ADGM [**?**]	18.44 ± 4.8	
SDGM [**?**]	8.11 ± 1.3	
Improved-GAN [**?**]	7.42 ± 0.65	
ALI [**?**]	7.05 ± 0.30	
Π-model [**?**]	5.43 ± 0.25	
Temporal ensembling [**?**]	5.12 ± 0.13	
FR-GAN with regularization	15.62 ± 0.16	500
FR-GAN	13.41 ± 0.20	500
FR-GAN with regularization	8.13 ± 0.22	1000
FR-GAN	7.20 ± 0.16	1000

4 Conclusion

This paper addresses the difficulty of traditional discriminator training modes in extracting more abstract and task-related robust features by proposing a feature recalibration generative adversarial network. To learn task-related features, we introduce unsupervised feature-level mean squared error loss for the discriminator under different model states, penalizing different outputs obtained from the same input in real samples. Meanwhile, the traditional discriminator structure composed of convolution and pooling modules is enhanced by introducing the SENet module in the discriminator' s intermediate layers, which automatically learns the importance of each feature channel to extract useful task-related features and suppress task-irrelevant features, thereby improving semi-supervised learning performance.

References

[1] Kingma D P, Rezende D J, Mohamed S, et al. Semi-supervised learning with deep generative models [EB/OL]. (2014-10-31). https://arxiv.org/pdf/1406.5298.pdf.

[2] Maaløe L, Sønderby C K, Sønderby S K, et al. Auxiliary deep generative models [EB/OL]. (2016-06-17). https://arxiv.org/pdf/1602.05473.pdf.

[3] Li Chongxuan, Zhu Jun, Zhang Bo. Max-margin deep generative models for (semi-) supervised learning [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2016, 3(9): 1-10.

[4] Rasmus A, Valpola H, Honkala M, et al. Semi-supervised learning with ladder networks [J]. Computer Science, 2015, 9: 1-9.

[5] Laine S, Aila T. Temporal ensembling for semi-supervised learning [EB/OL]. (2017-03-15). https://arxiv.org/abs/1610.02242.

[6] Miyato T, Maeda S, Koyama M, et al. Distributional smoothing with virtual adversarial training [J]. Computer Science, 2016, 5(2): 72-86.

[7] Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training GANs [EB/OL]. (2016-06-14). https://arxiv.org/pdf/1606.03498.pdf.

[8] Springenberg J T. Unsupervised and semi-supervised learning with categorical generative adversarial networks [J]. Computer Science, 2015, 6(5): 34-42.

[9] Li Chongxuan, Xu Kun, Zhu Jun, et al. Triple generative adversarial nets [EB/OL]. https://arxiv.org/abs/1703.02291.

[10] Dumoulin V, Belghazi I, Poole B, et al. Adversarially learned inference [EB/OL]. https://arxiv.org/abs/1606.00704.

[11] Jaderberg M, Simonyan K, Zisserman A. Spatial transformer networks [C]//Advances in Neural Information Processing Systems. 2015: 2017-2025.

[12] Jie Hu, Li Shen, Gang Sun. Squeeze-and-Excitation Networks [C]//Proc of Computer Vision and Pattern Recognition. 2017: 1-8.

[13] Zeiler M D, Krishnan D, Taylor G W, et al. Deconvolutional networks [C]//Computer Vision and Pattern Recognition, 2010: 2528-2535.

[14] Netzer Y, Wang Tao, Coates A, et al. Reading digits in natural images with unsupervised feature learning [C]//Proc of NIPS Workshop on Deep Learning and Unsupervised Feature Learning. 2011: 5-13.

[15] Bastien F, Lamblin P, Pascanu R, et al. Theano: new features and speed improvements [J]. Computer Science, 2012, 8(1): 51-63.

Note: Figure translations are in progress. See original paper for figures.