Abstract

We propose a novel regularization algorithm to train deep neural networks, in which data at training time is severely biased. Since a neural network efficiently learns data distribution, a network is likely to learn the bias information to categorize input data. It leads to poor performance at test time, if the bias is, in fact, irrelevant to the categorization. In this paper, we formulate a regularization loss based on mutual information between feature embedding and bias. Based on the idea of minimizing this mutual information, we propose an iterative algorithm to unlearn the bias information. We employ an additional network to predict the bias distribution and train the network adversarially against the feature embedding network. At the end of learning, the bias prediction network is not able to predict the bias not because it is poorly trained, but because the feature embedding network successfully unlearns the bias information. We also demonstrate quantitative and qualitative experimental results which show that our algorithm effectively removes the bias information from feature embedding.

1. Introduction

Machine learning algorithms and artificial intelligence have been used in wide ranging fields. The growing variety of applications has resulted in great demand for robust algorithms. The most ideal way to robustly train a neural network is to use suitable data free of bias. However great effort is often required to collect well-distributed data. Moreover, there is a lack of consensus as to what constitutes well-distributed data.

Apart from the philosophical problem, the data distribution significantly affects the characteristics of networks, as current deep learning based algorithms learn directly from the input data. If biased data is provided during training, the machine perceives the biased distribution as meaningful information. This perception is crucial because it weakens the robustness of the algorithm and unjust discrimination can be introduced.

A similar concept has been explored in the literature and is referred to as unknowns. In [2], the authors categorized unknowns as follows: known unknowns and unknown unknowns. The key criterion differentiating these categories is the confidence of the predictions made by the trained models. The unknown unknowns correspond to data points that the model’s predictions are wrong with high confidence, e.g. high softmax score, whereas the known unknowns represent mispredicted data points with low confidence. Known unknowns have better chance to be detected as the classifier’s confidence is low, whereas unknown unknowns are much difficult to detect as the classifier generates high confidence score.

In this study, the data bias we consider has a similar fla-
Our main contributions can be summarized as follows: First, we propose a novel regularization term, based on mutual information, to unlearn target bias from the given data. Second, we experimentally show that the proposed regularization term minimizes the detrimental effects of bias in the data. By removing information relating to the target bias from feature embedding, the network was able to learn more informative features for classification. In all experiments, networks trained with the proposed regularization loss showed performance improvements. Moreover, they achieved the best performance in the most experiments. Finally, we propose bias planting protocols for public datasets that can modify them to enhance suitability for bias removal problem.

2. Related Works

The existence of unknown unknowns was experimentally demonstrated by Attenberg et al. in [2]. The authors separated the decisions rendered by predictive models into four conceptual categories: known knowns, known unknowns, unknown knowns, and unknown unknowns. Subsequently, the authors developed and participated in a “beat the machine challenge”, which challenged the participants to manually find the unknown unknowns to fool the machine.

Several approaches for identifying unknown unknowns have been also proposed [11, 3]. Lakkaraju et al. [11] proposed an automatic algorithm using the explore-exploit strategy. Bansal and Weld proposed a coverage-based utility model that evaluates the coverage of discovered unknown unknowns [3]. These approaches rely on an oracle for a subset of test queries. Rather than relying on an oracle, Alvi et al. [1] proposed joint learning and unlearning method to remove bias from neural network embedding. To unlearn the bias, the authors applied confusion loss, which can be computed by calculating the cross-entropy of the output classifier and a uniform distribution.

As mentioned by Alvi et al. in the paper [1], the unsupervised domain adaptation (UDA) problem is closely related to the biased data problem. The UDA problem involves generalizing the network embedding over different domains [6, 22, 20]. The main difference between our problem and the UDA problem is that our problem does not assume the access to the target images and instead, we are aware of the description of the target bias.

Embracing the UDA problem, disentangling feature representation has been widely researched in the literature. The application of disentangled features has been explored in detail [23, 16]. The authors constructed new face images using a disentangled feature input, while preserving the original identities. Using generative adversarial network [7], more research to learn disentangled representation [4, 14, 21] have been proposed. In particular, Chen et al. proposed the InfoGAN [4] method, which learns and preserves semantic context without supervision.

These studies highlighted the importance of feature disentanglement, which is the first step in understanding the information contained within the feature. Inspired by various applications, we have attempted to remove certain information from the feature. In contrast to the InfoGan [4], we minimize the mutual information in order not to learn. However, removal of information is an antithetical concept...
to learning and is also referred to as unlearning. Although the concept itself is the complete opposite of learning, it can help learning algorithms. Herein, we describe an algorithm for removing target information and present experimental results and analysis to support the proposed algorithm.

3. Problem Statement

In this section, we formulate a novel regularization loss, which minimizes the undesirable effects of biased data, and describe the training procedure. The notations should be defined prior to introduction of the formulation. Unless specifically mentioned, all notation refers to the following terms hereafter. Assume we have an image \( x \in \mathcal{X} \) and corresponding label \( y_x \in \mathcal{Y} \). We define a set of bias, \( \mathcal{B} \), which contains every possible target bias that \( \mathcal{X} \) can possess. In Figure 1, \( \mathcal{B} \) is a set of possible colors, while \( \mathcal{Y} \) represents a set of digit classes. We also define a latent function \( b : \mathcal{X} \rightarrow \mathcal{B} \), where \( b(x) \) denotes the target bias of \( x \). We define random variables \( X \) and \( Y \) that have the value of \( x \) and \( y_x \) respectively.

The input image \( x \) is fed into the feature extraction network \( f : \mathcal{X} \rightarrow \mathbb{R}^K \), where \( K \) is the dimension of the feature embedded by \( f \). Subsequently, the extracted feature, \( f(x) \), is fed forward through the both label prediction network \( g : \mathbb{R}^K \rightarrow \mathcal{Y} \), and bias prediction network \( h : \mathbb{R}^K \rightarrow \mathcal{B} \). The parameters of each network are defined as \( \theta_f, \theta_g, \) and \( \theta_h \) with the subscripts indicating their specific network. Figure 2 describes the overall architecture of the neural networks. However, we do not explicitly designate a detailed architecture, since our regularization loss is applicable to various general network architectures.

3.1. Formulation

The objective of our work is to train a network that performs robustly with unbiased data during test time, even though the network is trained with biased data. The data bias has following characteristic:

\[
\mathcal{I}(b(X^{\text{train}}); Y) \gg \mathcal{I}(b(X); Y) \approx 0, \tag{1}
\]

where \( X^{\text{train}} \) denotes the random variable \( X \) sampled during the training procedure, and \( \mathcal{I}(\cdot; \cdot) \) denotes the mutual information. Biased training data results in the biased networks:

\[
\mathcal{I}(b(f(X)); g(f(X))) \gg 0. \tag{2}
\]

To this end, we add the mutual information to the objective function for training networks. We minimize the mutual information over \( f(x) \), instead of \( g(f(x)) \). It is adequate because the label prediction network, \( g \), takes \( f(x) \) as its input. From a standpoint of \( g \), the training data is not biased if the network \( f \) extracts no information of the target bias. In other words, extracted feature \( f(x) \) should contain no information of the target bias, \( b(x) \). Therefore, the training procedure is to optimize the following problem:

\[
\min_{\theta_f, \theta_g} \mathbb{E}_{x \sim p_X} [\mathcal{L}_c(y; g(f(x))) + \lambda \mathcal{I}(b(X); f(X))], \tag{3}
\]

where \( \mathcal{L}_c(\cdot, \cdot) \) represents the cross-entropy loss, and \( \lambda \) is a hyper-parameter to balance the terms. The \( \ell_2 \) regularization term for each parameter, also referred to weight decay regularization, was omitted for convenience. Weight decay was applied to all parameters in every experiment.

The mutual information in Eq. (3) can be equivalently expressed as follows:

\[
\mathcal{I}(b(X); f(X)) = H(b(X)) - H(b(X)| f(X)), \tag{4}
\]

where \( H(\cdot) \) and \( H(\cdot|\cdot) \) denote the marginal and conditional entropy, respectively. Since the marginal entropy of bias is constant that does not depend on \( \theta_h \) and \( \theta_g \), \( H(b(X)) \) can be omitted from the optimization problem. Eq. (4) is difficult to directly minimize as it requires the posterior distribution, \( P(b(X)| f(X)) \). Since it is not tractable in practice, minimizing the Eq. (4) is reformulated using an auxiliary distribution, \( Q \), with an additional equality constraint:

\[
\min_{\theta_f} \mathbb{E}_{x \sim p_X} [\mathbb{E}_{b \sim Q(b| f(x))} [\log Q(b| f(x))] ]
\]

s.t. \( Q(b(X)| f(X)) = P(b(X)| f(X)) \). \tag{5}

The benefit of using the \( Q \) distribution is that we can directly calculate the objective function. Therefore, we can train the feature extraction network, \( f \), under the equality constraint.

3.2. Training Procedure

Before we describe the training procedure, we need to further interpret the equality constraint. In Eq. (5), the equality constraint contradicts the main purpose of the regularization; minimizing the objective function removes bias
information from \( f(X) \), whereas the constraint implies that the bias is still predictable from \( f(X) \). To resolve this contradiction, we changed the optimization problem into a minimax game \([7]\). We relax the Eq. (5) using Lagrangian contradiction, we changed the optimization problem into a

\[
L_{M1} = E \tilde{x} \sim P_X(\cdot) [E_{\tilde{b} \sim Q(\cdot|f(\tilde{x}))}[\log Q(b|f(\tilde{x}))]] - \mu D_{KL}(P(b(X)|f(X))||Q(b(X)|f(X))),
\]

where \( \mu \) is a Lagrangian multiplier and \( D \) denotes the KL-divergence. Note that we will train network \( h \), so that the KL-divergence is minimized, i.e. \( h \) tries to maximize \( L_{M1} \).

Similar to the method proposed by Chen et al. \([4]\), we parametrize the auxiliary distribution, \( Q \), as the bias prediction network, \( h \). Although the posterior distribution, \( P(b(X)|f(X)) \), is not tractable, the bias prediction network, \( h \), is expected to be trained to approximate \( P(b(X)|f(X)) \), if we train the network with \( P(b(X)) \) as the label with stochastic gradient descent optimizer. Therefore, we can replace the KL-divergence of Eq. (6) with the cross-entropy loss between \( b(X) \) and \( h(f(X)) \). The reformulation of \( L_{M1} \) is

\[
L_{M1}(\theta_f, \theta_h) = E \tilde{x} \sim P_X(\cdot) [E_{\tilde{b} \sim h(b(\tilde{x}))}(f(\tilde{x}))][\log h(b|f(\tilde{x}))]] - \mu L_{c}(b(\tilde{x}), h(f(\tilde{x}))).
\]

With Eq. (7), we let the networks, \( f \) and \( h \), to play the minimax game. We train \( h \) to correctly predict the bias, \( b(x) \), from its feature embedding, \( f(x) \). Simultaneously, we train \( f \) to minimize the conditional entropy. Together with the main classification problem, the minimax game is formulated as follows:

\[
\min_{\theta_f} \max_{\theta_h} \mathbb{E}_{\tilde{x} \sim P_X(\cdot)} [L_c(y, g(f(\tilde{x}))) + \lambda L_{M1}(\theta_f, \theta_h)].
\]

In practice, the deep neural networks, \( f \), \( g \) and \( h \), are trained with both adversarial strategy \([7, 4]\) and gradient reversal technique \([6]\). Early in learning, \( g \circ f \) are rapidly trained to classify the label using the bias information because the gradient signal to minimize \( L_{M1}(\theta_f, \theta_h) \) is almost a random signal with poor bias prediction network, \( h \). Then \( h \) learns to predict the bias, and \( f \) begins to learn how to extract feature embedding independent of the bias. At the end of the training, \( h \) regresses to the poor performing network not because the bias prediction network, \( h \), diverges, but because \( f \) unlearns the bias, so the feature embedding, \( f(X) \), does not have enough information to predict the target bias.

4. Dataset

Most existing benchmarks are designed to evaluate a specific problem. The collectors often split the dataset into train/test sets exquisitely. However, their efforts to maintain the train/test split to obtain an identical distribution obscures our experiment. Thus, we intentionally planted bias to well-balanced public benchmarks to determine whether our algorithm could unlearn the bias.

4.1. Colored MNIST

The MNIST dataset \([12]\) is a widely used handwritten digit database used for image recognition. It contains grayscale images from ten digit categories. We planted a color bias into the MNIST dataset. To synthesize the color bias, we selected ten distinct colors and assigned them to each digit category as their mean color. Then, for each training image, we randomly sampled a color from the normal distribution of the corresponding mean color and provided variance, and colorized the image. Since the variance of the normal distribution is a parameter that can be controlled, the amount of the color bias in the data can be adjusted. For each test image, we randomly choose a mean color among the ten pre-defined colors and followed the same colorization protocol as for the training images. Each sub-datasets are denoted as follows:

- Train-\( \sigma^2 \): Train images with colors sampled with \( \sigma^2 \)
- Test-\( \sigma^2 \): Test images with colors sampled with \( \sigma^2 \)

Since the digits in the test sets are colored with random mean colors, the Test-\( \sigma^2 \) sets are unbiased. We varied \( \sigma^2 \) from 0.02 to 0.05 with a 0.005 interval. Smaller values of \( \sigma^2 \) indicate more bias in the set. Thus, Train-0.02 is the most biased set, whereas Train-0.05 is the least biased.

Figure 3 (a) shows samples from the colored MNIST, where the images in the training set show that the color and digit class are highly correlated. The color of the digit contains sufficient information to categorize the digits in the training set, but it is insufficient for the images in the test set. Recognizing the color would rather disrupt the digit categorization. Therefore, the color information must be removed from the feature embedding.

4.2. Dogs and Cats

We evaluated our algorithm with the dogs and cats database, developed by kaggle \([10]\). The original database is a set of 25K images of dogs and cats for training and 12,500 images for testing. Similar to \([11]\), we manually categorized the data according to the color of the animal: bright, dark, and other. Subsequently, we split the images into three subsets.

- Train-biased 1 (TB1): bright dogs and dark cats.
Figure 3. Examples of datasets with intentionally planted bias. (a) We modified the MNIST data [12] to plant color bias to train images. A mean color has been designated for each class, so a classifier can easily predict the digit with color. (b) TB1 is a set of bright dogs and dark cats, whereas TB2 contains dark dogs and bright cats. Similar to the colored MNIST, a classifier can predict whether an image is dog or cat with its color. (c) IMDB face dataset contains age and gender labels. EB1 and EB2 are different on the correlation between age and gender. Predicting age enables an algorithm to predict gender. We did not plant bias to the test set of each dataset to verify whether an algorithm is capable of predicting the label independent of the bias.

- Train-biased 2 (TB1): dark dogs and bright cats.
- Test set: All 12,500 images from the original test set.

The images categorized as other are images featuring white cats with dark brown stripes or dalmatians. They were not used in our experiments due to their ambiguity. In turn, TB1 and TB2 contain 10,047 and 6,738 images respectively. The constructed dogs and cats dataset is shown in Figure 3 (b), with each set containing a color bias. Unlike TB1 and TB2, the test set does not contain color bias.

On the other hand, the ground truth labels for test images are not accessible, while the data is originally for competition [10]. Therefore, we trained an oracle network (ResNet-18 [8]) with all 25K training images. For the test set, we measured the performance based on the result from the oracle network. We presumed that the oracle network could accurately predict the label since it is a simple classification task.

4.3. IMDB Face

The IMDB face dataset [18] is a publicly available face image dataset. It contains 460,723 face images from 20,284 celebrities along with information regarding their age and gender. Each image in the IMDB face dataset is a cropped facial image. As mentioned in [18, 1], the provided label contains significant noise. To filter out misannotated images, we used pretrained networks [13] on Adience benchmark [5] designed for age and gender classification. Using the pretrained networks, we estimated the age and gender for all the individuals shown in the images in the IMDB face dataset. We then collected images where the both age and gender labels match with the estimation. From this, we obtained a cleaned dataset with 112,340 face images, and the detailed cleaning procedure is described in the supplementary material.

Similar to the protocol from [1], we classified the cleaned IMDB images into three biased subsets. We first withheld 20% of the cleaned IMDB images as the test set, then split the rest of the images as follows:

- Extreme bias 1 (EB1): women aged 0-29, men aged 40+
- Extreme bias 2 (EB2): women aged 40+, men aged 0-29
- Test set: 20% of the cleaned images aged 0-29 or 40+

As a result, EB1 and EB2 contain 36,004 and 16,800 facial images...
images respectively, and the test set contains 13129 images. Figure 3 (c) shows that both EB1 and EB2 are biased with respect to the age. Although it is not as clear as the color bias in Figure 3 (a) and (b), EB1 consists of younger female and older male celebrities, whereas EB2 consists of younger male and older female celebrities.

5. Experiments

5.1. Implementation

In the following experiments, we removed three types of target bias: color, age, and gender. The age and gender labels were provided in IMDB face dataset, therefore $\mathcal{L}_{MI}(\theta_f, \theta_h)$ was optimized with supervision. On the other hand, the color bias was removed via self-supervision. To construct color labels, we first sub-sampled the images by factor of 4. Consequently, the dynamic range of color, 0-255, was quantized into eight even levels.

For the network architecture, we used ResNet-18 [8] for real images and plain network with four convolution layers for the colored MNIST experiments. The network architectures correspond to the parametrization of $g \circ f$. In the case we used ResNet-18, $g$ was implemented as two residual blocks on the top, while $f$ represents the rest. For plain network for colored MNIST, both $g$ and $f$ consist of two convolution layers. ResNet-18 was pretrained with ImageNet data [19] except for the last fully connected layer. We implemented $h$ with two convolution layers for color bias and single fully connected layer for gender and age bias. Every convolution layer is followed by batch normalization [9] and ReLU activation layers.

To train the networks, a stochastic gradient descent optimizer was used with a learning rate of 0.001 and momentum of 0.9. The hyper-parameters, $\lambda$ and $\mu$, are fixed as 0.1 and 1, respectively. Although it is not described in Eq. 8, the adaptation parameter of the gradient reversal layer [6] was fixed as 0.1 for all experiments. Each experiment was conducted using PyTorch [15] and repeated five times. All the evaluation results were averaged to be presented in this paper.

5.2. Results

We compare our training algorithm with other methods that can be used for this task. The performance of the algorithms mentioned in this section were re-implemented based on the literature.

Colored MNIST. The amount of bias in the data was controlled by adjusting the value of $\sigma^2$. A network was trained for each $\sigma^2$ value from 0.02 to 0.05 and was evaluated with the corresponding test set with the same $\sigma^2$. Since a color for each image was sampled with a given $\sigma^2$, smaller $\sigma^2$ implies severer color bias. Figure 4 shows the evaluation results of the colored MNIST. The baseline model represents a network trained without additional regularization and the baseline performance can roughly be used as an indication of training data bias. The algorithm denoted as “BlindEye” represents a network trained with confusion loss [1] instead of $\mathcal{L}_{MI}(\theta_f, \theta_h)$. The other algorithm, denoted as “Gray”, represents a network trained with grayscale images and it was also tested with grayscale images. For the given color biased data, we converted the color digits into grayscale. Conversion into grayscale is a trivial approach that can be used to mitigate the color bias. We presume that the conversion into grayscale does not reduce the information significantly since the MNIST dataset was originally provided in grayscale.

The results of our proposed algorithm outperformed the BlindEye [1] and baseline model with all values of $\sigma^2$. Notably, we achieved similar performance as the model trained and tested with grayscale images. Since we converted images in both training and test time, the network is hardly biased. In most experiments, our model performed slightly better than the gray algorithm, suggesting that our regulation algorithm can effectively remove the target bias and encourage a network to extract more informative features.

To analyze the effect of the bias and proposed algorithm,
we re-colored the test images. We sampled with the same protocol, but with fixed mean color, once assigned to one of the ten digit classes. Figure 5 shows the confusion matrices drawn by the baseline and our models with the re-colored test images. The digits illustrated in the top row denotes the mean colors and their corresponding digit class in training set. For example, the first digit, red zero, signifies the confusion matrices below are drawn by test images colored reddish regardless of their true label. It also stands for a fact that every digit of category zero in training data is colored reddish.

In Figure 5, the matrices of the baseline show vertical patterns, some of which are shared, such as digits 1 and 3. The mean color for class 1 is teal; in RGB space it is (0, 149, 182) and is called bondi blue. This indicates that the baseline network is biased to the color of digit. As observed from the figure, the confusion matrices drawn by our algorithm (bottom row) show that the color bias was removed.

**Dogs and Cats.** Table 1 presents the evaluation results, where the baseline networks perform admirably, considering the complexity of the task due to the pretrained parameters. As mentioned in [17], neural networks prefer to categorize images based on shape rather than color. This encourages the baseline network to learn shapes, but the evaluation results presented in Table 1 imply that the networks remain biased without regularization.

Similar to the experiment on the colored MNIST, simplest approach for removing the color bias is to convert the images into grayscale. Unlike the MNIST dataset, conversion would remove a significant amount of information. Although the networks for grayscale images performed better than the baseline, Table 1 shows that the networks remain biased to color. This is likely because of the criterion that was used to implant the color bias. Since the original dataset is categorized into bright and dark, the converted images contain a bias in terms of brightness. In the colored MNIST experiment, the brightness hardly leads to bias since there are ten classes with various brightness values.

We used gradient reversal layer (GRL) [6] and adversarial training strategy [4, 7] as components of our optimization process. To analyze the effect of each component, we ablated the GRL from our algorithm. We also trained networks with both confusion loss [1] and GRL, since they can be used in conjunction with each other. Although the GRL was originally proposed to solve unsupervised domain adaptation problem [6], Table 1 shows that it is beneficial for bias removal. Together with either confusion loss or \( L_{MIL}(\theta_f, \theta_h) \), we obtained the performance improvements. Furthermore, GRL alone notably improved the performance suggesting that GRL itself is able to remove bias.

Figure 6 shows the qualitative effect of our proposed regularization. The prediction results of the baseline networks are constant whether the query image is cat or dog if the colors are identical. If a network is trained with TB1, the network predicts a dark image to be a cat and a bright image to be a dog. If another network is trained with TB2, the network predicts a bright image to be a cat and a dark image to be a dog. This implies that the baseline networks are biased to color. On the other hand, networks trained with our proposed algorithm successfully classified the query images independent of their color. In particular, Figure 6 (c) and (f) were identically predicted by the baseline networks depending on their color. After removing the color information from the feature embedding, the images were correctly categorized according to their appearance.

**IMDB face.** For the IMDB face dataset, we conducted two experiments; one to train the networks to classify age independent of gender, and one to train the networks to classify gender independent of age. Table 2 shows the evaluation results from both experiments. The networks were trained
Figure 6. Qualitative results of dogs and cats dataset. The oracle model was trained with not only both TB1 and TB2, but also with images we categorized as other color. For test images, prediction results of the oracle model were considered as their true labels. The stacked bar charts below the figures visualize prediction results by each model. The baseline models tend to predict depends on the color, whereas our model ignores the color information while prediction.

Figure 7. Qualitative results of gender classification with IMDB face dataset. Same as the Figure 6, the stacked bar charts represent the prediction results. They show that the baseline models are bias to the age. On the other hand, the networks trained with proposed algorithm predict the gender independent of their age.

with either EB1 or EB2 and since they are extremely biased, the baseline networks are also biased. By removing the target bias information from the feature embedding, overall performances are improved. On the other hand, considering that gender classification is a two class problem, where random guessing achieves 50% accuracy, the networks perform poorly on gender classification. Although Table 2 shows that the performance improves after removing the target bias from the feature embedding, the performance improvement achieved using our algorithm is marginal compared to previous experiments with other datasets. We presume that this is because of the correlation between age and gender. In the case of color bias, the bias itself is completely independent of the categories. In other words, an effort to unlearn the bias is purely beneficial for digit categorization. Thus, removing color bias from feature embedding improved the performance significantly because the network is able to focus on learning shape feature. Unlike the color bias, age and gender are not completely independent features. Therefore, removing bias information from feature embedding would not completely beneficial. This suggests that a deep understanding of the specific data bias must be preceded by the removal of bias.

Figure 7 shows the qualitative effect of regularization on the gender classification task. Young, mid-age, and old individuals both male and female are presented. Similar to Figure 6, it implies that the baseline networks are biased toward age. The baseline network trained with EB1 predicted both young male and female images (Figure 7 (a) and (d)) as female with high confidence. Meanwhile, the network trained with EB2 predicted the same images as the exact opposite gender with high confidence. Upon removal of age bias, the networks were trained to correctly predict the gender.

6. Conclusion

In this paper, we propose a novel regularization term to train deep neural networks using biased data. The core idea of using mutual information is inspired by InfoGan [4]. In the context of the inspiring approach, we rather minimize the mutual information in order not to learn. By letting networks to play minimax game, networks learn to categorize, while unlearn the bias. The experimental results showed...
that the networks trained with proposed regularization can extract bias-independent feature embedding, achieving the best performance in the most of the experiments. Furthermore, our model performed better than “Gray” model which was trained with unbiased data, indicating the feature embedding becomes even more informative. To conclude, we have demonstrated in this paper that proposed regularization improves the performance of neural networks trained with biased data. We expect this study to expand the usage of various data and to contribute to the field of feature disentanglement.

References

[1] M. Alvi, A. Zisserman, and C. Nellaker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. arXiv preprint arXiv:1809.02169, 2018. 2, 5, 6, 7
[2] J. Attenberg, P. Ipeirotis, and F. Provost. Beat the machine: Challenging humans to find a predictive model’s unknown unknowns. Journal of Data and Information Quality (JDQI), 6(1):1, 2015. 1, 2
[3] G. Bansal and D. S. Weld. A coverage-based utility model for identifying unknown unknowns. In Proc. of AAAI, 2018. 2
[4] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. CoRR, abs/1606.03657, 2016. 2, 4, 7, 8
[5] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. IEEE Transactions on Information Forensics and Security, 9(12):2170–2179, Dec 2014. 5
[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. The Journal of Machine Learning Research, 17(1):2096–2030, 2016. 2, 4, 6, 7
[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014. 2, 4, 7
[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015. 3, 5, 6
[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR, abs/1502.03167, 2015. 6
[10] Kaggle. Dogs vs. cats, 2013. 4, 5
[11] H. Lakkaraju, E. Kamar, R. Caruana, and E. Horvitz. Discovering blind spots of predictive models: Representations and policies for guided exploration. CoRR, abs/1510.02192, 2015. 2
[12] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. 4, 5
[13] G. Levi and T. Hassnner. Age and gender classification using convolutional neural networks. In 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 34–42, June 2015. 5
[14] Y. Liu, Z. Wang, H. Jin, and I. Wassell. Multi-task adversarial network for disentangled feature learning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. 2
[15] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In NIPS-W, 2017. 6
[16] X. Peng, X. Yu, K. Sohn, D. N. Metaxas, and M. Chandraker. Reconstruction for feature disentanglement in pose-invariant face recognition. CoRR, abs/1702.03041, 2017. 2
[17] S. Ritter, D. G. Barrett, A. Santoro, and M. M. Botvinick. Cognitive psychology for deep neural networks: A shape bias case study. arXiv preprint arXiv:1706.08606, 2017. 7
[18] R. Rothe, R. Timofte, and L. V. Gool. Dex: Deep expectation of apparent age from a single image. In IEEE International Conference on Computer Vision Workshops (ICCVW), December 2015. 5
[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015. 6
[20] O. Sener, H. O. Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 2110–2118. Curran Associates, Inc., 2016. 2
[21] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In Proceeding of IEEE Computer Vision and Pattern Recognition, Honolulu, HI, July 2017. 2
[22] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. CoRR, abs/1510.02192, 2015. 2
[23] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim. Rotating your face using multi-task deep neural network. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 676–684, June 2015. 2