Bias Mitigation in Fine-tuning Pre-trained Models for Enhanced Fairness and Efficiency

Yixuan Zhang[1] , **Feng Zhou**[2] ,
[1]Hangzhou Dianzi University
[2]Remin University of China
yixuan.zhang@hdu.edu.cn, feng.zhou@ruc.edu.cn

Abstract

Fine-tuning pre-trained models is a widely employed technique in numerous real-world applications. However, fine-tuning these models on new tasks can lead to unfair outcomes. This is due to the absence of generalization guarantees for fairness properties, regardless of whether the original pre-trained model was developed with fairness considerations. To tackle this issue, we introduce an efficient and robust fine-tuning framework specifically designed to mitigate biases in new tasks. Our empirical analysis shows that the parameters in the pre-trained model that affect predictions for different demographic groups are different, so based on this observation, we employ a transfer learning strategy that neutralizes the importance of these influential weights, determined using Fisher information across demographic groups. Additionally, we integrate this weight importance neutralization strategy with a matrix factorization technique, which provides a low-rank approximation of the weight matrix using fewer parameters, reducing the computational demands. Experiments on multiple pre-trained models and new tasks demonstrate the effectiveness of our method.

1 Introduction

Over recent decades, automated decision-making systems have been applied in extensive application in numerous fields, including medicine [Kim *et al.*, 2015], finance [Khandani *et al.*, 2010], criminology [Dressel and Farid, 2018], etc. The broad adoption of machine learning techniques raises concerns regarding its potential to exhibit unfair behavior, as these systems are data-driven and can inherit the biases present in their input data. This replication of bias by the models can result in biased decisions that unfairly discriminate, posing significant risks to individuals and society [Bird *et al.*, 2016]. Consequently, it is crucial to develop machine learning algorithms that are free from discrimination against specific demographic groups.

Existing literature indicates two primary strategies for achieving fairness in machine learning models. The first approach involves integrating fairness constraints directly into the training phase, often referred to as in-processing. This method modifies the objective functions to account for fairness considerations. Various studies have explored this avenue by incorporating fairness constraints into the optimization process, aiming to mitigate biases in the learned model [Bilal Zafar *et al.*, 2016; Agarwal *et al.*, 2018; Kamishima *et al.*, 2012]. In contrast, the second strategy involves learning fair representations as a pre-processing step. This method focuses on creating unbiased and fair representations of the input data before employing conventional machine learning techniques. These fair representations serve as the foundation for subsequent learning tasks, ensuring that the underlying data used for training the models is inherently unbiased [Zemel *et al.*, 2013; Calmon *et al.*, 2017; Zhang *et al.*, 2023].

The preceding methods, whether in-processing or pre-processing, require constructing a new model from scratch for a specific task. However, in practical scenarios, retraining a model from the ground up for every new task is computationally intensive. Therefore, a more common approach is to fine-tune existing pre-trained models. For instance, with the prevalence of large models today, when adapting to new task domains, it is impractical to train these massive models from scratch. Instead, we often opt to fine-tune them by, for example, freezing the feature extraction part and only training the final linear layer. However, fine-tuning pre-trained models on new tasks can result in unpredictable unfair outcomes. This unpredictability remains even when the pre-trained model was trained with fairness objectives, primarily because there is a lack of assurance regarding the fairness property's generalizability [Kamishima *et al.*, 2012; Oneto *et al.*, 2020].

We employ a transfer learning strategy to address the specified limitations, focusing on neutralizing the importance of influential weights to reduce bias. The intuition is clear: from our experiments, it is evident that distinct weights in the pre-trained model have a substantial impact on predictions across different demographic groups. Therefore, our objective is to balance the impact of these influential weights, making it challenging to differentiate group information from the weights, thus mitigating bias. Due to the enormous parameter count in large pre-trained models, even fine-tuning just the linear layer requires a considerable amount of time. Therefore, to further improve efficiency, we enhance the method by

approximating the weights in the linear layer with fewer parameters using singular value decomposition (SVD) [Golub and Reinsch, 1971]. Our method consists of three steps: (1) We assess the importance of linear-layer weights by employing Fisher information across various demographic groups, subsequently neutralizing these importance scores. (2) Utilizing these neutralized importance scores, we conduct a weighted SVD of the linear-layer weight matrix. (3) We replace the original linear layer with the low-rank layers obtained from SVD and fine-tune them for the new task.

Specifically, we make the following contributions: (1) We propose a weight importance neutralization strategy to mitigate bias for fine-tuning on new tasks, taking into account the influence of each parameter on prediction across different demographic groups. (2) To enhance the fine-tuning efficiency, we use SVD to create a low-rank approximation of the linear-layer weight matrix, which has fewer parameters. (3) Our empirical analysis shows that even with a fair pre-trained model, fine-tuning on new tasks can result in severe bias, and our method can effectively address this issue.

The structure of this paper is as follows: Section 2 reviews prior work in the domain of fairness-aware learning, model fine-tuning, and low-rank approximation. Section 3 details the methodology we propose. In Section 4, we conduct the experimental comparison of our method with various baselines and leading-edge techniques across three real-world datasets. Lastly, Section 5 summarizes this work and outlines potential avenues for future research.

2 Related Work

In this section, we briefly review fairness-aware learning methods, model fine-tuning, and low-rank approximation literature which are most relevant to ours.

2.1 Bias Mitigation

Numerous recent studies have demonstrated that modern machine learning models exhibit biases against certain demographic groups [Buolamwini and Gebru, 2018]. Existing bias mitigation methods can be generally grouped into two categories. The first focuses on the in-processing stage to remove discrimination, such as modifying the objective functions [Roh *et al.*, 2020; Agarwal *et al.*, 2018], employing adversarial learning [Zhang *et al.*, 2018] or using boosting techniques [Iosifidis and Ntoutsi, 2019]. This kind of approach aims for a fair classifier that ensures predictions are independent of sensitive variables. Typically, the methods in this category require annotations of sensitive variables so that the fairness criteria can be applied in the training process.

The second family of bias mitigation methods is based on representation learning. It involves developing different neural network architectures with modified learning strategies to learn representations that remove the undesirable relationship between sensitive variables and non-sensitive variables in the raw data [Creager *et al.*, 2019; Dwork *et al.*, 2011; Zemel *et al.*, 2013]. In this setting, the learned representations preserve information that is useful for prediction while removing the dependency on sensitive variables.

2.2 Adaptation and Fine-tuning

Adaptation and fine-tuning from pre-trained models is a widely used technique in various real-world applications. Previously, comprehensive works used full fine-tuning, which involves initializing the model with pre-trained parameters across all layers, and then subsequently training the model on specific downstream tasks [Liu *et al.*, 2019; Tayaranian Hosseini *et al.*, 2023]. However, with the growth in the number of parameters of deep learning models (with billions of parameters), it becomes challenging to adapt these models to downstream tasks, therefore, parameter-efficient fine-tuning has been proposed to reduce the computational demands, which introduces new, trainable parameters for task-specific fine-tuning [He *et al.*, 2022; Lin *et al.*, 2020]. Beyond the methods mentioned above, partial fine-tuning is another strategy, focusing on updating only a selected subgroup of pre-trained parameters that are crucial for downstream tasks [Xu *et al.*, 2021; Fu *et al.*, 2023]. The recent popular methods in the low-rank decomposition family are categorized as reparameterized fine-tuning, which utilizes low-rank decomposition to reduce the number of trainable parameters, particularly useful for pre-trained weights [Hu *et al.*, 2022; Aghajanyan *et al.*, 2021]. In this work, we adopt a hybrid fine-tuning strategy, combining the benefits of partial fine-tuning with the low-rank decomposition method.

2.3 Low-rank Approximation

Fine-tuning solely the linear layer of large pre-trained models can still demand significant computational resources due to their extensive parameter count. To improve efficiency during the fine-tuning phase, strategies like low-rank decomposition have been introduced. These techniques aim to closely approximate the weight matrix in the linear layer with a more compact version [Hu *et al.*, 2022; Jaderberg *et al.*, 2014]. Recently, SVD has been widely applied in low-rank approximation. This approach is geared towards compressing the weight parameters to improve efficiency, as noted in several studies [Ben Noach and Goldberg, 2020; Acharya *et al.*, 2018]. By employing SVD and similar techniques, the goal is to significantly minimize the reconstruction error of the weight matrix by using fewer parameters, thereby achieving a more efficient model without substantially sacrificing performance.

3 Method

In this section, we first introduce the notations used in Section 3.1. Then, we present empirical findings that illustrate how fine-tuning a pre-trained model on new tasks can lead to unfair results, as detailed in Section 3.2. After this analysis, we introduce our weight importance neutralization technique in Section 3.3, specifically formulated to facilitate fair fine-tuning on new tasks.

3.1 Notations

We focus on the supervised learning task in the context of binary classification. The training set of a new task is defined as $\mathcal{D} = \{(x_i, y_i, s_i)\}_{i=1}^{N}$, where $x_i \in \mathcal{X} \subset \mathbb{R}^D$ denotes the high-dimensional non-sensitive feature, $y_i \in \mathcal{Y} = \{-1, 1\}$

denotes the binary ground-truth label, and $s_i \in \mathcal{S} = \{1, 2\}$ denotes the binary sensitive feature, such as gender or race. $\mathcal{D}_{S=s} = \{(x_i, y_i, s_i = s)\}_{i=1}^{N_s}$ denotes a subset of the dataset, which consists of samples sharing the same sensitive feature. The pre-trained model is expressed as $f(x; \theta)$ with θ parameterizing the model. The pre-trained model $f(x; \theta)$ can generally be divided into two sequential parts: a feature extractor that transforms the raw data into a representation, and a classification head (usually a linear layer) that converts the representation into output probabilities. The predicted class is determined by $\hat{y} = \arg\max f(x; \theta)$. The pre-trained model is fine-tuned on the dataset $\mathcal{D}$ to adapt to a new task. In this process, the feature extractor is frozen, and only the classification head is fine-tuned. Our primary objective is to alleviate discrimination during fine-tuning.

3.2 Fine-tuning Introduces Bias

This section starts with an exploration of the discrimination caused by fine-tuning the pre-trained models on new tasks. To this end, we pre-train an MLP on the Adult Income dataset [Kohavi, 1996], using 60% of the data for pre-training and reserving the rest 40% as the new task. Due to the inherent presence of discrimination in the Adult Income dataset, the pre-trained model obtained in this way is considered unfair, denoted as f_B. As a control group, we also pre-train an MLP with the same data but using the in-processing fairness-aware learning approach [Bilal Zafar *et al.*, 2015]. In this approach, we incorporate a demographic parity constraint into the objective function. We treat this pre-trained model that considers fairness as a fair one, which is denoted as f_F. Then we fine-tune both f_F and f_B on the new task.

For ease of visualization, we apply the principal component analysis (PCA) to the representation obtained from f_F and f_B after fine-tuning, reducing it to 2 dimensions. Fig. 1 illustrates the distribution of the representations after dimensionality reduction for samples predicted as positive in the "Male" and "Female" groups for f_F and f_B. It is evident that the representations obtained from f_B display a different pattern across different demographic groups. While for f_F, the pattern is less pronounced than that of f_B, yet there is still a noticeable division into two groups in the plot.

These findings demonstrate that bias can be introduced when a pre-trained model is fine-tuned on a new dataset. This phenomenon occurs regardless of whether the original pre-trained model is developed with fairness considerations or not. This discovery is significant, as it suggests that many large language models (LLM) obtained from fine-tuning may also encounter fairness issues, such as ChatGPT fine-tuned from GPT-3 and Sparrow fine-tuned from Chinchilla.

3.3 Weight Importance Neutralization

Based on the empirical findings in Section 3.2, this section presents our *Weight Importance Neutralization* method. This approach effectively reduces the bias inherited from fine-tuning the pre-trained model on new tasks. We first address fundamental questions: What is weight importance? Why do we consider it as the starting point of our method?

Weight Importance via Fisher Information. The concept of weight importance is derived from Fisher information [Pascanu and Bengio, 2014], which is a metric used to evaluate how crucial certain parameters are to the model's prediction. Similar to Section 3.2, we conduct experiments on the Adult Income dataset with an MLP. As illustrated in Fig. 3, we plot the diagonal entries of the Fisher information matrix for all parameters (20 in total) of the final linear layer of the model. This is done separately for the model trained by $\mathcal{D}_{S=1}$ (represents "Female" group) or $\mathcal{D}_{S=2}$ (represents "Male" group). From Fig. 3, it is clear that the influential weights for prediction across two demographic groups vary. Taking inspiration from this observation, we employ Fisher information to quantify the impact of parameters on the performance of the model trained by $\mathcal{D}_{S=1}$ or $\mathcal{D}_{S=2}$, and our goal is to mitigate this difference by neutralizing it.

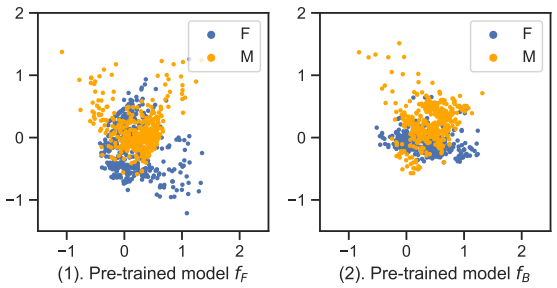

Figure 1: Principal component analysis on the representations from the linear layer after fine-tuning the pre-trained models. Blue points represent the "Female" group, while orange points represent the "Male" group.

In practice, the exact form of Fisher information is normally intractable due to the marginalization over the space of data, so we use the empirical version to approximate the exact form. The empirical version is given by:

$$\begin{aligned} I_\theta &= \mathbb{E}\left[\left(\frac{\partial}{\partial\theta}\log p(\mathcal{D} \mid \theta)\right)^2\right] \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\partial}{\partial\theta}\mathcal{L}(x_i, s_i, y_i; \theta)\right)^2 = \hat{I}_\theta, \end{aligned} \tag{1}$$

where $\mathcal{L}$ is the objective function of the corresponding model.

Weight Importance Neutralization. Because incorporating the full Fisher information matrix into our weighting strategy would be computationally intensive, we follow the same assumption setting in [Hsu *et al.*, 2022] where each row of the weight matrix is assumed to share the same importance score. Specifically, we explicitly denote the weight and bias parameters from θ as W and b, respectively. Instead of calculating the entire matrix, we define the estimated Fisher information as $\hat{I}_{W_i} = \sum_j \hat{I}_{W_{ij}}$. Subsequently, we assume $\hat{I} = \text{diag}(\sqrt{\hat{I}_{W_1}}, \cdots, \sqrt{\hat{I}_{W_d}})$. This allows us to more effectively calculate the importance of each weight entry in relation to different tasks, denoted as $\hat{I}_{S=1}$ and $\hat{I}_{S=2}$ for $\mathcal{D}_{S=1}$ and $\mathcal{D}_{S=2}$, respectively. Finally, the neutralized Fisher information is expressed as:

$$\hat{I}_N = \frac{1}{2}(\hat{I}_{S=1} + \hat{I}_{S=2}). \tag{2}$$