DEBIASING NEURAL NETWORKS USING DIFFERENTIABLE CLASSIFICATION PARITY PROXIES

Ričards Marcinkevičs, Ece Ozkan and Julia E. Vogt
Department of Computer Science, ETH Zürich
{ricardsm, ece.oezkanelsen, julia.vogt}@inf.ethz.ch

ABSTRACT

Due to growing concerns about demographic disparities and discrimination resulting from algorithmic and model-based decision-making, recent research has focused on mitigating biases against already disadvantaged or marginalised groups in classification models. From the perspective of classification parity, the two commonest metrics for assessing fairness are statistical parity and equality of opportunity. Current approaches to debiasing in classification either require the knowledge of the protected attribute before or during training or are entirely agnostic to the model class and parameters. This work considers differentiable proxy functions for statistical parity and equality of opportunity and introduces two novel debiasing techniques for neural network classifiers based on fine-tuning and pruning an already-trained network. As opposed to the prior work leveraging adversarial training, the proposed methods are simple yet effective and can be readily applied *post hoc*. Our experimental results encouragingly suggest that these approaches successfully debias fully connected neural networks trained on tabular data and often outperform model-agnostic post-processing methods.

1 INTRODUCTION

Ethical considerations are pertinent to ML systems involved in high-stakes decisions. For instance, an ICU patient monitoring and management model trained on a dataset containing few patients from minority groups might suffer from under- or over-detection of events in these groups, subsequently leading to alarm fatigue among medical staff and disparate patient outcomes (Rajkomar et al., 2018). Motivated by such concerns, researchers have provided many solutions for adjusting models' outputs and directly incorporating fairness into the learning process (Kearns, 2017). This paper focuses on debiasing neural networks from the perspective of classification parity (Corbett-Davies & Goel, 2018): a classifier is said to be fair if some derivative of its confusion matrix, for instance, the true positive rate (TPR), is even across the categories of the protected attribute, such as race or gender.

Prior works on debiasing classifiers have mostly assumed that the protected attribute is known either before or at the time of training or have taken a completely model-agnostic approach. By contrast, we consider the intra-processing setting (Savani et al., 2020), where an already-trained network needs to be debiased and the debiasing procedure may access and edit the model's parameters. In this setting, we seek a neural network debiased w.r.t. statistical parity (Besse et al., 2021) or equality of opportunity (Hardt et al., 2016). The latter scenario emerges when potential biases are unknown, unexplored, or cannot be foreseen at the training time. For instance, consider deploying a predictive neural network model in several hospitals with different demographics, e.g. Zech et al. (2018) explore such a multi-centre setup for chest X-ray classification — it is more practical to debias such a model based on the local needs and data than retrain from scratch. Importantly, as we will see, intra-processing is more flexible than popular post-processing, which has a similar motivation.

Contributions Our main contributions are as follows: (*i*) we consider differentiable proxy functions for statistical parity and equality of opportunity and establish their correspondence to the covariance between the decision boundary of a neural network and the protected attribute; (*ii*) we introduce simple yet effective intra-processing debiasing procedures based on minimising the proxy functions via fine-tuning and pruning an already-trained neural network; (*iii*) we conduct comprehensive comparison among the proposed and well-established post-processing approaches.

2 BACKGROUND

Notation Throughout the paper, we will assume given training, validation, and test datasets $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, a_i)\}_i = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}} \cup \mathcal{D}_{\text{test}}$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ is a feature vector, $y_i \in \{0, 1\}$ is the label, and $a_i \in \{0, 1\}$ is the protected attribute. Attribute a_i may be present among the features in $\boldsymbol{x}_i$ or may be completely exogenous. Let $f_{\boldsymbol{\theta}}(\cdot)$ denote a multilayer perceptron (MLP) with weights $\boldsymbol{\theta} = \left\{\boldsymbol{W}^{\text{in}}, \boldsymbol{W}^1, ..., \boldsymbol{W}^L, \boldsymbol{W}^{\text{out}}\right\}$ trained on data points $\{(\boldsymbol{x}_i, y_i)\}_i$ from $\mathcal{D}_{\text{train}}$. For $1 \leq l \leq L$ and $\boldsymbol{x} \in \mathbb{R}^p$, let $\boldsymbol{h}^l(\boldsymbol{x}) = \sigma\left(\boldsymbol{W}^l \boldsymbol{h}^{l-1}(\boldsymbol{x})\right)$ be the activation vector of the l-th layer, where $\sigma(\cdot)$ is an activation function and $\boldsymbol{h}^0(\boldsymbol{x}) = \sigma\left(\boldsymbol{W}^{\text{in}} \boldsymbol{x}\right)$. The output of the MLP is given by $\hat{y} = \text{sigmoid}\left(\boldsymbol{W}^{\text{out}} \boldsymbol{h}^L(\boldsymbol{x})\right)$.

Classification Parity Many criteria for the fairness of machine learning models have been considered so far (Corbett-Davies & Goel, 2018). Arguably, the two most common and practical classification parity metrics are statistical parity and equality of opportunity. Statistical parity difference (SPD) (Savani et al., 2020; Besse et al., 2021)is defined as the difference between the probabilities of positive outcomes across the groups of the protected attribute: $\text{SPD} = \mathbb{P}_{\boldsymbol{X},A}\left(f_{\boldsymbol{\theta}}(\boldsymbol{X}) = 1 | A = 0\right) - \mathbb{P}_{\boldsymbol{X},A}\left(f_{\boldsymbol{\theta}}(\boldsymbol{X}) = 1 | A = 1\right)$. On the other hand, equal opportunity difference (EOD) (Hardt et al., 2016; Savani et al., 2020) quantifies the discrepancy between TPRs within the groups: $\text{EOD} = \mathbb{P}_{\boldsymbol{X},Y,A}\left(f_{\boldsymbol{\theta}}(\boldsymbol{X}) = 1 | Y = 1, A = 0\right) - \mathbb{P}_{\boldsymbol{X},Y,A}\left(f_{\boldsymbol{\theta}}(\boldsymbol{X}) = 1 | Y = 1, A = 1\right)$.

Debiasing Minimisation of the SPD or EOD above is a solvable technical problem, and many debiasing algorithms have been proposed in this context, e.g. by Hardt et al. (2016), Zafar et al. (2017), and Zhang et al. (2018). Bellamy et al. (2018) and Savani et al. (2020) provide a practical taxonomy of the debiasing methods: (*i*) pre-processing algorithms usually reweigh or transform original data, obfuscating protected variables or attenuating group disparities (Kamiran & Calders, 2011; Zemel et al., 2013; Calmon et al., 2017; Celis et al., 2020); (*ii*) in-processing methods incorporate debiasing explicitly into learning, e.g. using an adversarial loss or regularisation (Kamishima et al., 2012; Zafar et al., 2017; Zhang et al., 2018; Reimers et al., 2021); (*iii*) post-processing approaches treat the biased model as a black-box and merely edit its predictions (Kamiran et al., 2012; Hardt et al., 2016; Pleiss et al., 2017); (*iv*) intra-processing techniques, are inspired by fine-tuning and achieve parity by changing the model's parameters *post hoc* (Savani et al., 2020).

3 METHODS

Classification Parity Proxies Some methods for debiasing neural networks resort to adversarial training (Zhang et al., 2018; Reimers et al., 2021), wherein a discriminator network is trained in parallel to predict the protected attribute based on an intermediate or output layer in the network. By contrast, we seek to minimise SPD or EOD *directly* using differentiable proxy functions. Given sets $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, $\mathcal{Y} = \{y_i\}_{i=1}^N$, and $\mathcal{A} = \{a_i\}_{i=1}^N$, the proxies for the SPD and EOD are

$$\tilde{\mu}_{\text{SPD}}\left(f_{\boldsymbol{\theta}}, \mathcal{X}, \mathcal{Y}, \mathcal{A}\right) = \frac{\sum_{i=1}^N f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right)\left(1 - a_i\right)}{\sum_{i=1}^N 1 - a_i} - \frac{\sum_{i=1}^N f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right) a_i}{\sum_{i=1}^N a_i}, \tag{1}$$

$$\tilde{\mu}_{\text{EOD}}\left(f_{\boldsymbol{\theta}}, \mathcal{X}, \mathcal{Y}, \mathcal{A}\right) = \frac{\sum_{i=1}^N f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right)\left(1 - a_i\right) y_i}{\sum_{i=1}^N \left(1 - a_i\right) y_i} - \frac{\sum_{i=1}^N f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right) a_i y_i}{\sum_{i=1}^N a_i y_i}. \tag{2}$$

Notably, Equation 1 is similar to the objective function considered by Zafar et al. (2017) in the context of linear classifiers. Moreover, we show that it corresponds to the empirical estimate of the covariance between A and $f_{\boldsymbol{\theta}}\left(\boldsymbol{X}\right)$. Similarly, we show that Equation 2 corresponds to the empirical estimate of the conditional covariance between A and $f_{\boldsymbol{\theta}}\left(\boldsymbol{X}\right)$ given that $Y = 1$ (see Appendix B).

Constrained Objective In practice, many debiasing techniques often lead to a decrease in the overall predictive performance (Reimers et al., 2021). The purpose of a debiasing algorithm should be to reduce bias μ, e.g. given by the SPD or EOD, without sacrificing performance ρ, e.g. balanced accuracy (Brodersen et al., 2010). To this end, we consider the maximisation of the bias-constrained objective (Savani et al., 2020):

$$\varphi_{\rho,\mu,\varepsilon}\left(f_{\boldsymbol{\theta}}, \boldsymbol{X}, Y, A\right) = \begin{cases} \rho\left(f_{\boldsymbol{\theta}}, \boldsymbol{X}, Y\right), & \text{if } \left|\mu\left(f_{\boldsymbol{\theta}}, \boldsymbol{X}, Y, A\right)\right| < \varepsilon \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $\varepsilon > 0$ is an upper bound on the bias. Below we present two novel intra-processing approaches to maximising this constrained objective. These methods complement the previous work by Savani et al. (2020) (for an extended discussion, refer to Appendix A).

Pruning Pruning refers to the procedure reducing the effective number of parameters in a model. There has been renewed interest in neural network pruning (Cheng et al., 2017; Blalock et al., 2020), mainly for model compression and reductions in computational and memory complexity and energy consumption. Different from existing work, we investigate the use of pruning in the context of mitigating bias. In particular, we introduce a procedure for pruning individual units, or neurons, in a neural network based on their contributions to classification disparity. Inspired by the literature on interpreting individual neurons and attribution measures for them (Leino et al., 2018; Dhamdhere et al., 2019; Bau et al., 2020), we propose the following gradient-based statistic for the influence of the j-th unit in the l-th layer on a differentiable bias measure $\tilde{\mu}$, e.g. specified by Equation 1 or 2:

$$S_{l,j} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \tilde{\mu} \left(f_{\boldsymbol{\theta}}, \mathcal{X}, \mathcal{Y}, \mathcal{A} \right)}{\partial h_j^l(\boldsymbol{x}_i)}, \tag{4}$$

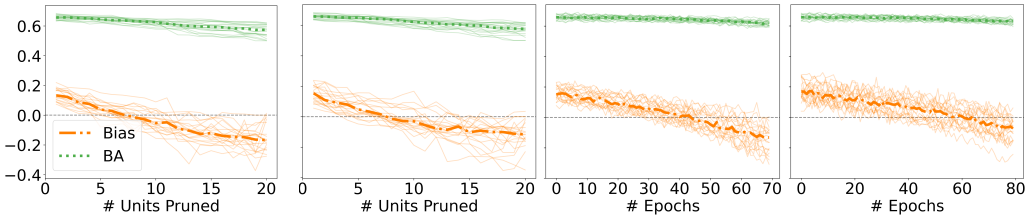
where $h_j^l(\boldsymbol{x}_i)$ denotes unit's activation at data point $\boldsymbol{x}_i$. In practice, partial derivatives such as above can be computed efficiently using automatic differentiation, e.g. PyTorch's Autograd module (Paszke et al., 2017). Our pruning procedure is quite simple (see Algorithm 1 in Appendix C): (*i*) for layer $1 \leq l \leq L$, evaluate influence $S_{l,j}$ (see Equation 4) for each unit j; (*ii*) prune most influential units by setting all outgoing weights to 0's, e.g. for the j-th unit in the l-th layer this amounts to the assignment $\boldsymbol{W}_{j,\cdot}^l \leftarrow \mathbf{0}$; (*iii*) evaluate the bias-constrained objective (see Equation 3) for the pruned network on held-out validation data $\mathcal{D}_{\text{valid}}$; (*iv*) recompute $S_{l,j}$ for the pruned network and repeat steps (*ii*)–(*iv*). In the end, an optimal sparsity level is chosen based on the constrained objective evaluated on held-out data. In summary, the procedure above greedily removes individual units in the intermediate layers of the neural network step-by-step based on the criterion given by Equation 4. It returns a pruned network that has a maximal bias-constrained objective. We will see below that, in practice, it often allows making few changes to the classifier without retraining from scratch and sacrificing the predictive performance while dramatically reducing the classification disparity.

Bias Gradient Descent/Ascent Since the proxies given by Equations 1 and 2 are differentiable w.r.t. parameters $\boldsymbol{\theta}$, one could minimise them directly using gradient descent or ascent, depending on the sign of the bias. Therefore, another approach we consider is essentially fine-tuning the classifier $f_{\boldsymbol{\theta}}(\cdot)$ for a few epochs with a small learning rate, for instance, using mini-batch gradient descent and Equation 1 or 2 as a loss function (see Algorithm 2 in Appendix C). This method is markedly similar to the adversarial debiasing by Zhang et al. (2018), whose discriminator is applied to the network's output. However, in our case, we investigate performing gradient descent/ascent on the differentiable bias proxies *after* the network was trained.

4 RESULTS

We conducted experiments on a few standard benchmarking datasets publicly available in IBM's AI Fairness 360 open source toolkit (Bellamy et al., 2018) and MIMIC database (Johnson et al., 2016). In particular, we ran debiasing for the following datasets: Adult, Bank, COMPAS, and MIMIC-III with "*sex*", "*age*", "*race*", and "*insurance*" as protected attributes, respectively. Across all datasets, we used the same fully connected architecture and training scheme for the classifier $f_{\boldsymbol{\theta}}(\cdot)$, similar to Savani et al. (2020). We compared our proposed pruning (PRUNING) and bias gradient descent/ascent (BIAS GD/A) to several baseline debiasing methods, including equalised odds (EQ. ODDS) (Hardt et al., 2016), reject option classification (ROC) (Kamiran et al., 2012), and random perturbation (RANDOM) (Savani et al., 2020). We assessed bias using the SPD and EOD and predictive performance w.r.t. balanced accuracy (BA) for all methods. We used the bias-constrained objective (CO) with an upper bound of $\varepsilon = 0.05$ for method comparison (see Equation 3). Appendix D contains further details on the experimental setup.

Figure 1 shows changes in the EOD and SPD during pruning and gradient descent/ascent on COMPAS data. Encouragingly, both proposed methods achieve classification parity while not affecting the balanced accuracy of the classifier significantly. Table 1 contains bias-constrained objectives attained by different debiasing methods. Compared to the post-processing techniques and



(a) PRUNING, SPD (b) PRUNING, EOD (c) BIAS GD/A, SPD (d) BIAS GD/A, EOD

Figure 1: Changes in the **bias**, given by the SPD (a, c) and EOD (b, d), and **balanced accuracy** of the neural network during pruning (a, b) and bias gradient descent/ascent (c, d). The results were obtained on COMPAS from 20 train-test splits. **Bold** lines correspond to the median across 20 seeds.

Table 1: Bias-constrained objectives (at $\varepsilon = 0.05$, see Equation 3) attained after debiasing. STANDARD refers to the original neural network. For each dataset, debiasing was run twice: for the SPD and EOD separately. The objectives are reported as medians followed by 25 and 75% quantiles across 20 train-test splits. Best results are shown in **bold**, second-best – in *italic*.

Bias Measure	Method	Adult: *Sex*	Bank: *Age*	COMPAS: *Race*	MIMIC-III: *Insurance*
SPD	STANDARD	0.00; [0.00, 0.00]	0.00; [0.00, 0.00]	0.00; [0.00, 0.00]	0.00; [0.00, 0.00]
	RANDOM	0.59; [0.59, 0.60]	0.52; [0.00, 0.55]	0.00; [0.00, 0.00]	0.67; [0.66, 0.68]
	ROC	**0.78; [0.00, 0.80]**	0.00; [0.00, 0.56]	0.50; [0.50, 0.50]	0.66; [0.00, 0.67]
	EQ. ODDS	0.00; [0.00, 0.00]	0.00; [0.00, 0.69]	0.59; [0.00, 0.60]	0.57; [0.56, 0.58]
	PRUNING	0.54; [0.52, 0.57]	*0.83; [0.80, 0.85]*	*0.62; [0.41, 0.64]*	*0.70; [0.69, 0.71]*
	BIAS GD/A	*0.67; [0.64, 0.69]*	**0.85; [0.00, 0.87]**	**0.63; [0.46, 0.64]**	**0.73; [0.73, 0.74]**
EOD	STANDARD	0.00; [0.00, 0.00]	**0.86; [0.00, 0.87]**	0.00; [0.00, 0.00]	0.37; [0.00, 0.75]
	RANDOM	0.00; [0.00, 0.00]	**0.86; [0.00, 0.87]**	0.00; [0.00, 0.00]	*0.74; [0.00, 0.76]*
	ROC	**0.81; [0.00, 0.82]**	**0.86; [0.00, 0.87]**	0.50; [0.50, 0.50]	0.72; [0.00, 0.74]
	EQ. ODDS	*0.72; [0.53, 0.74]*	*0.00; [0.00, 0.68]*	*0.59; [0.00, 0.60]*	0.57; [0.55, 0.57]
	PRUNING	**0.81; [0.79, 0.81]**	**0.86; [0.00, 0.87]**	0.56; [0.00, 0.62]	**0.75; [0.55, 0.75]**
	BIAS GD/A	**0.81; [0.00, 0.82]**	**0.86; [0.00, 0.87]**	**0.62; [0.00, 0.64]**	**0.75; [0.55, 0.75]**

random perturbation, on most datasets, pruning and bias gradient descent/ascent are successful at mitigating biases, tending to sacrifice less accuracy, and having narrower interquartile ranges for the constrained objective. On average, pruning performs slightly worse than gradient descent/ascent and has larger variability across seeds (see Figure 1). An intuitive explanation could be that pruning, compared to gradient descent/ascent, explores a relatively limited number of debiased network weight configurations, particularly for smaller architectures. Interestingly, on Adult dataset, both procedures drastically reduce the BA of the classifier and perform worse than ROC: we attribute this to the general sensitivity of intra-processing methods (Savani et al., 2020) to initial conditions (see Appendix E for further results). In summary, the experiments suggest that the proposed intra-processing approaches effectively reduce bias when it is present and offer improved performance over model-agnostic techniques, even in relatively small and simple neural network architectures and tabular datasets.

5 CONCLUSION

This paper considered differentiable proxy functions for statistical parity and equality of opportunity. We proposed two novel intra-processing debiasing procedures based on neural network pruning and fine-tuning that utilise these proxies. Our preliminary experimental results on tabular data with fully connected neural network architectures are promising and indicate the viability of the proposed methods, especially compared to model-agnostic post-processing.

Future Work Current experimental setup is limited to relatively simple datasets and architectures: in our future work, we plan to apply proposed methods to medical imaging data and convolutional neural networks. It would be interesting to investigate strategies for pruning beyond the gradient-based influence and consider a more general setting with multiple classes and protected attribute categories. Last but not least, our methods should be compared to stronger baselines, e.g. adversarial in- or intra-processing and debiasing based on zeroth-order optimisation methods.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Mona Azadkia, Dr. Alexander Marx, and Imant Daunhawer for valuable discussions and feedback. Ričards Marcinkevičs was supported by the SNSF grant #320038189096.

REFERENCES

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Gradient-based attribution methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 169–191. Springer International Publishing, 2019.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020.

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, 2018. arXiv:1810.01943.

Philippe Besse, Eustasio del Barrio, Paula Gordaliza, Jean-Michel Loubes, and Laurent Risser. A survey of bias in machine learning through the prism of statistical parity. *The American Statistician*, pp. 1–11, 2021.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? In *Proceedings of Machine Learning and Systems*, volume 2, pp. 129–146, 2020.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *20th International Conference on Pattern Recognition*. IEEE, 2010.

Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

L. Elisa Celis, Vijay Keswani, and Nisheeth Vishnoi. Data preprocessing to mitigate bias: A maximum entropy based approach. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1349–1359. PMLR, 2020.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks, 2017. arXiv:1710.09282.

Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning, 2018. arXiv:1808.00023.

Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. How important is a neuron? In *International Conference on Learning Representations*, 2019.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring Adult: New datasets for fair machine learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1993.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1), 2016.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2011.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*, pp. 924–929, 2012.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases*, pp. 35–50. Springer Berlin Heidelberg, 2012.

Michael Kearns. Fair algorithms for machine learning. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pp. 1. Association for Computing Machinery, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

Ron Kohavi. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 202–207. AAAI Press, 1996.

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the COMPAS recidivism algorithm. `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`, 2016. Accessed: 2020.11.02.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1990.

Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks. In *IEEE International Test Conference (ITC)*. IEEE, 2018.

Wei-Yin Loh, Luxi Cao, and Peigen Zhou. Subgroup identification for precision medicine: A comparative review of 13 methods. *WIREs Data Mining and Knowledge Discovery*, 9(5), 2019.

Chuizheng Meng, Loc Trinh, Nan Xu, and Yan Liu. MIMIC-IF: Interpretability and fairness evaluation of deep learning models on MIMIC-IV dataset, 2021. arXiv:2102.06761.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR*, 2017.

Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2501–2508, 2020.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch, 2017. URL `https://openreview.net/forum?id=BJJsrmfCZ`. NIPS 2017 Autodiff Workshop.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. On fairness and calibration. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5684–5693. Curran Associates Inc., 2017.

Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmarking deep learning models on large healthcare datasets. *Journal of Biomedical Informatics*, 83:112–134, 2018.

Tai Le Quy, Arjun Roy, Vasileios Iosifidis, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning, 2021. arXiv:2110.00530.

Alvin Rajkomar, Michaela Hardt, Michael D. Howell, Greg Corrado, and Marshall H. Chin. Ensuring fairness in machine learning to advance health equity. *Annals of Internal Medicine*, 169(12): 866–872, 2018.

Christian Reimers, Paul Bodesheim, Jakob Runge, and Joachim Denzler. Conditional adversarial debiasing: Towards learning unbiased classifiers from biased data. In *Pattern Recognition*, pp. 48–62. Springer International Publishing, 2021.

Yash Savani, Colin White, and Naveen Sundar Govindarajulu. Intra-processing methods for debiasing neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2798–2810. Curran Associates, Inc., 2020.

Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P. Gummadi. Fairness Constraints: Mechanisms for Fair Classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pp. 962–970. PMLR, 2017.

John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLOS Medicine*, 15(11):e1002683, 2018.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pp. 325–333. PMLR, 2013.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2018.

A Related Work

Below we provide an extended overview of the related work and make further remarks on the connections between the proposed methods (see Section 3) and other debiasing techniques.

Pruning In the context of neural networks, parameter pruning usually refers to the removal of irrelevant weights or entire structural elements (Cheng et al., 2017), e.g. of filters in convolutional neural networks. Early works on pruning neural networks, such as optimal brain damage (LeCun et al., 1990) and optimal brain surgeon (Hassibi & Stork, 1993), leveraged criteria based on the second derivative of the error function to prune unimportant weights throughout the training process. Several modern techniques focus on pruning entire structures (Wen et al., 2016; Molchanov et al., 2017; He et al., 2017), e.g. convolutional filters or channels. However, the main principle remains the same: parameters are pruned based on some criterion, and the network is subsequently fine-tuned by backpropagation, if necessary.

Role of Individual Units in Neural Networks Several lines of work have investigated the importance and interpretation of *individual* neurons within deep neural network models, by contrast to the previous research on attribution, which primarily examined input-output relationships (Ancona et al., 2019). For instance, Bau et al. (2020) observe the emergence of single-unit object detectors whose activations are correlated with high-level concepts essential for the classification task. Leino et al. (2018); Dhamdhere et al. (2019); Srinivas & Fleuret (2019); Nam et al. (2020) introduce new attribution measures that quantify the influence of individual neurons.

Further Remarks Herein, we draw comparisons between the most closely related work and the proposed pruning and bias gradient descent/ascent techniques. To the best of our knowledge, neural network pruning has never been considered from the perspective of debiasing for fairness. The instigation of this line of work is one of the contributions of the current paper. Similar differentiable proxies and the constrained optimisation setting have been investigated by Zafar et al. (2017) w.r.t. the SPD. However, their analysis is limited to linear classifiers, such as logistic regression and SVM. On the other hand, minimisation of the SPD or EOD has been implemented by Zhang et al. (2018) via adversarial training wherein a discriminator is applied to the output of the classifier. As mentioned, adversarial debiasing requires knowing the protected attribute *during* training. In contrast, our bias gradient descent/ascent procedure is applied *post hoc* to a classifier that has been trained without accounting for the protected variables. Last but not least, there exist similarities between bias gradient descent/ascent and adversarial fine-tuning by Savani et al. (2020). Notably, Savani et al. (2020) apply a discriminator to a hidden layer, whereas in our case, proxies are defined based on the network's output, thus, leading to a different loss function. Moreover, our method has fewer hyperparameters, and the loss functions we utilise are directly related to the decision boundary covariance (see Appendix B).

B Decision Boundary Covariance

Lemma 1. *For* $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, $\mathcal{Y} = \{y_i\}_{i=1}^N$, *and* $\mathcal{A} = \{a_i\}_{i=1}^N$ *and some classifier* $f_{\boldsymbol{\theta}}(\cdot)$, $-\tilde{\mu}_{\mathrm{SPD}}\left(f_{\boldsymbol{\theta}},\, \mathcal{X},\, \mathcal{Y},\, \mathcal{A}\right) \propto \widehat{\mathrm{Cov}}\left(A,\, f_{\boldsymbol{\theta}}\left(\boldsymbol{X}\right)\right)$.

Proof. Recall that the covariance is given by

$$\mathrm{Cov}\left(A,\, f_{\boldsymbol{\theta}}\left(\boldsymbol{X}\right)\right) = \mathbb{E}\left[A f_{\boldsymbol{\theta}}\left(\boldsymbol{X}\right)\right] - \mathbb{E}\left[A\right]\mathbb{E}\left[f_{\boldsymbol{\theta}}(\boldsymbol{X})\right].$$

Let $K = \sum_{i=1}^N a_i$ and $\overline{f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)} = \frac{1}{N}\sum_i^N f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right)$, consider an empirical estimate

$$\widehat{\mathrm{Cov}}\left(A,\, f_{\boldsymbol{\theta}}\left(\boldsymbol{X}\right)\right) = \frac{1}{N}\sum_{i=1}^N f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right) a_i - \frac{K}{N^2}\sum_{i=1}^N f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right) = \frac{1}{N}\sum_{i=1}^N f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right) a_i - \frac{K}{N}\overline{f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)}. \quad (5)$$

Observe that

$$-\tilde{\mu}_{\text{SPD}} = \frac{\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, a_i}{\sum_{i=1}^{N} a_i} - \frac{\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\,(1-a_i)}{\sum_{i=1}^{N}(1-a_i)} = \frac{1}{K}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, a_i - \frac{N}{N-K}\overline{f_{\boldsymbol{\theta}}(\boldsymbol{x})} - \frac{1}{N-K}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, a_i = \frac{N}{K(N-K)}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, a_i - \frac{NK}{K(N-K)}\overline{f_{\boldsymbol{\theta}}(\boldsymbol{x})}. \tag{6}$$

Note that (5) $\propto$ (6) by a factor of $\frac{N^2}{K(N-K)}$, constant in $\boldsymbol{\theta}$. □

Lemma 2. *For $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, $\mathcal{Y} = \{y_i\}_{i=1}^N$, and $\mathcal{A} = \{a_i\}_{i=1}^N$ and some classifier $f_{\boldsymbol{\theta}}(\cdot)$, $-\tilde{\mu}_{\text{EOD}}(f_{\boldsymbol{\theta}}, \mathcal{X}, \mathcal{Y}, \mathcal{A}) \propto \widehat{\text{Cov}}(A,\, f_{\boldsymbol{\theta}}(\boldsymbol{X}) \mid Y = 1)$.*

Proof. Recall that, by the law of total covariance,

$$\text{Cov}(A,\, f_{\boldsymbol{\theta}}(\boldsymbol{X}) \mid Y=1) = \mathbb{E}\left[(A - \mathbb{E}[A \mid Y=1])\,(f_{\boldsymbol{\theta}}(\boldsymbol{X}) - \mathbb{E}[f_{\boldsymbol{\theta}}(\boldsymbol{X}) \mid Y=1]) \mid Y=1\right] = \mathbb{E}\left[A f_{\boldsymbol{\theta}}(\boldsymbol{X}) \mid Y=1\right] - \mathbb{E}[A \mid Y=1]\,\mathbb{E}\left[f_{\boldsymbol{\theta}}(\boldsymbol{X}) \mid Y=1\right].$$

Let $M = \sum_{i=1}^N y_i$, $R = \sum_{i=1}^N a_i y_i$, and $\overline{f_{\boldsymbol{\theta}}(\boldsymbol{x})} = \frac{1}{N}\sum_i^N f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$, consider an empirical estimate

$$\widehat{\text{Cov}}(A,\, f_{\boldsymbol{\theta}}(\boldsymbol{X}) \mid Y=1) = \frac{\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, a_i y_i}{\sum_{i=1}^{N} y_i} - \frac{\sum_{i=1}^{N} a_i y_i}{\sum_{i=1}^{N} y_i} \cdot \frac{\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i}{\sum_{i=1}^{N} y_i} = \frac{1}{M}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, a_i y_i - \frac{R}{M^2}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i. \tag{7}$$

Observe that

$$-\tilde{\mu}_{\text{EOD}} = \frac{\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i a_i}{\sum_{i=1}^{N} y_i a_i} - \frac{\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i (1-a_i)}{\sum_{i=1}^{N} y_i (1-a_i)} = \frac{1}{R}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i a_i - \frac{1}{M-R}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i - \frac{1}{M-R}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i a_i = \frac{M}{R(M-R)}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i a_i - \frac{1}{M-R}\sum_{i=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\, y_i. \tag{8}$$

Note that (7) $\propto$ (8) by a factor of $\frac{M^2}{R(M-R)}$, constant in $\boldsymbol{\theta}$. □

C ALGORITHMS

Algorithms 1 and 2 contain pseudocode for pruning and bias gradient descent/ascent procedures, respectively, described in Section 3. Note that the order in which units are pruned in Algorithm 1 and the weight update direction in Algorithm 2 depend on the sign of the initial bias, i.e. whether the bias needs to be driven down or up towards 0. Both algorithms have a tuning parameter – margin $0 \leq \gamma < \varepsilon$. During experiments, we have observed that for some datasets, both algorithms are sensitive to the choice of γ. Furthermore, bias gradient descent/ascent has additional hyperparameters, namely, learning rate $\eta > 0$, which in practice, should be chosen sufficiently small, mini-batch size $M \geq 1$, and a maximum number of fine-tuning epochs $E \geq 1$. Observe that pruning is performed only on hidden layers of the network. However, Algorithm 1 can be readily extended to prune inputs as well. Although Algorithm 2 is based on the mini-batch gradient descent, other optimisation procedures can be adopted, e.g. batch or stochastic gradient descent.

Algorithm 1: Pruning for Debiasing Neural Networks

Input: Training set $\mathcal{D}_{\text{train}}$; held-out validation set $\mathcal{D}_{\text{valid}} = \{(\boldsymbol{x}_i, y_i, a_i)\}_{i=1}^{N_{\text{valid}}}$; fully connected feedforward neural network $f_{\boldsymbol{\theta}}(\cdot)$ with weights $\boldsymbol{\theta} = \left\{\boldsymbol{W}^{\text{in}}, \boldsymbol{W}^1, ..., \boldsymbol{W}^L, \boldsymbol{W}^{\text{out}}\right\}$ trained on $\mathcal{D}_{\text{train}}$; predictive performance measure ρ; bias measure μ; differentiable bias proxy $\tilde{\mu}$; upper bound $\varepsilon > 0$; margin $0 \leq \gamma < \varepsilon$; number of steps $B \geq 1$

Output: Pruned and debiased network $f_{\tilde{\boldsymbol{\theta}}}(\cdot)$ with weights $\tilde{\boldsymbol{\theta}} = \left\{\tilde{\boldsymbol{W}}^{\text{in}}, \tilde{\boldsymbol{W}}^1, ..., \tilde{\boldsymbol{W}}^L, \tilde{\boldsymbol{W}}^{\text{out}}\right\}$

Let $\mu_0 \leftarrow \mu\left(f_{\boldsymbol{\theta}}, \{\boldsymbol{x}_i\}_{i=1}^{N_{\text{valid}}}, \{y_i\}_{i=1}^{N_{\text{valid}}}, \{a_i\}_{i=1}^{N_{\text{valid}}}\right)$, where $(\boldsymbol{x}_i, y_i, a_i) \in \mathcal{D}_{\text{valid}}$

Initialise $\tilde{\boldsymbol{W}}^{\text{in}} \leftarrow \boldsymbol{W}^{\text{in}}$; $\tilde{\boldsymbol{W}}^{\text{out}} \leftarrow \boldsymbol{W}^{\text{out}}$; $\tilde{\boldsymbol{W}}^l \leftarrow \boldsymbol{W}^l$ for $1 \leq l \leq L$

Based on the data $\mathcal{D}_{\text{train}}$ and proxy $\tilde{\mu}$, evaluate influence $S_{l,j}$ (see Equation 4) for every unit j in layer $1 \leq l \leq L$

for $b = 0$ *to* $B - 1$ **do**

  Let $\tau_b \leftarrow q_{1-1/B}\left(\{\text{sgn}\left(\mu_0\right) S_{l,j}\}\right)$, where $q_\alpha(\cdot)$ denotes the empirical α-quantile

  For all j and $1 \leq l \leq L$, set $\tilde{\boldsymbol{W}}^l_{j,\cdot} \leftarrow \boldsymbol{0}$ if $\text{sgn}\left(\mu_0\right) S_{l,j} > \tau_b$

  Let $\phi_b \leftarrow \varphi_{\rho,\mu,\varepsilon-\gamma}\left(f_{\tilde{\boldsymbol{\theta}}}, \{\boldsymbol{x}_i\}_{i=1}^{N_{\text{valid}}}, \{y_i\}_{i=1}^{N_{\text{valid}}}, \{a_i\}_{i=1}^{N_{\text{valid}}}\right)$, where $(\boldsymbol{x}_i, y_i, a_i) \in \mathcal{D}_{\text{valid}}$ (see Equation 3)

  Let $\tilde{\boldsymbol{\theta}}_b \leftarrow \tilde{\boldsymbol{\theta}}$

  Recompute influence $S_{l,j}$ for the pruned network $f_{\tilde{\boldsymbol{\theta}}}(\cdot)$

end

Let $b^* \leftarrow \arg\max_{0 \leq b \leq B-1} \phi_b$

return $f_{\tilde{\boldsymbol{\theta}}_{b^*}}(\cdot)$

Algorithm 2: Bias Gradient Descent/Ascent

Input: Training set $\mathcal{D}_{\text{train}}$; held-out validation set $\mathcal{D}_{\text{valid}} = \{(\boldsymbol{x}_i, y_i, a_i)\}_{i=1}^{N_{\text{valid}}}$; fully connected feedforward neural network $f_{\boldsymbol{\theta}}(\cdot)$ with weights $\boldsymbol{\theta} = \left\{\boldsymbol{W}^{\text{in}}, \boldsymbol{W}^1, ..., \boldsymbol{W}^L, \boldsymbol{W}^{\text{out}}\right\}$ trained on $\mathcal{D}_{\text{train}}$; predictive performance measure ρ; bias measure μ; differentiable bias proxy $\tilde{\mu}$; upper bound $\varepsilon > 0$; margin $0 \leq \gamma < \varepsilon$; learning rate $\eta > 0$; number of epochs $E \geq 1$; mini-batch size $M \geq 1$

Output: Fine-tuned and debiased network $f_{\tilde{\boldsymbol{\theta}}}(\cdot)$ with weights $\tilde{\boldsymbol{\theta}} = \left\{\tilde{\boldsymbol{W}}^{\text{in}}, \tilde{\boldsymbol{W}}^1, ..., \tilde{\boldsymbol{W}}^L, \tilde{\boldsymbol{W}}^{\text{out}}\right\}$

Let $\mu_0 \leftarrow \mu\left(f_{\boldsymbol{\theta}}, \{\boldsymbol{x}_i\}_{i=1}^{N_{\text{valid}}}, \{y_i\}_{i=1}^{N_{\text{valid}}}, \{a_i\}_{i=1}^{N_{\text{valid}}}\right)$, where $(\boldsymbol{x}_i, y_i, a_i) \in \mathcal{D}_{\text{valid}}$

Initialise $\tilde{\boldsymbol{W}}^{\text{in}} \leftarrow \boldsymbol{W}^{\text{in}}$; $\tilde{\boldsymbol{W}}^{\text{out}} \leftarrow \boldsymbol{W}^{\text{out}}$; $\tilde{\boldsymbol{W}}^l \leftarrow \boldsymbol{W}^l$ for $1 \leq l \leq L$

for $e = 0$ *to* $E - 1$ **do**

  Draw mini-batch $\mathcal{B} = \{(\boldsymbol{x}_i, y_i, a_i)\}_{i=1}^{M}$ without replacement, s.t. $\mathcal{B} \subseteq \mathcal{D}_{\text{train}}$

  Evaluate bias $\tilde{\mu}_e \leftarrow \tilde{\mu}\left(f_{\tilde{\boldsymbol{\theta}}}, \{\boldsymbol{x}_i\}_{i=1}^{M}, \{y_i\}_{i=1}^{M}, \{a_i\}_{i=1}^{M}\right)$, where $(\boldsymbol{x}_i, y_i, a_i) \in \mathcal{B}$

  Update $\tilde{\boldsymbol{W}} \leftarrow \tilde{\boldsymbol{W}} - \text{sgn}\left(\mu_0\right) \eta \nabla_{\tilde{\boldsymbol{W}}} \tilde{\mu}_e$ for all $\tilde{\boldsymbol{W}} \in \tilde{\boldsymbol{\theta}}$

  Let $\phi_e \leftarrow \varphi_{\rho,\mu,\varepsilon-\gamma}\left(f_{\tilde{\boldsymbol{\theta}}}, \{\boldsymbol{x}_i\}_{i=1}^{N_{\text{valid}}}, \{y_i\}_{i=1}^{N_{\text{valid}}}, \{a_i\}_{i=1}^{N_{\text{valid}}}\right)$, where $(\boldsymbol{x}_i, y_i, a_i) \in \mathcal{D}_{\text{valid}}$ (see Equation 3)

  Let $\tilde{\boldsymbol{\theta}}_e \leftarrow \tilde{\boldsymbol{\theta}}$

end

Let $e^* \leftarrow \arg\max_{0 \leq e \leq E-1} \phi_e$

return $f_{\tilde{\boldsymbol{\theta}}_{e^*}}(\cdot)$

D EXPERIMENTAL SETUP

This appendix outlines the setup of the experiments described in Section 4. Appendix D.1 briefly describes benchmarking datasets used; Appendix D.2 details the neural network architecture and training scheme used throughout all debiasing experiments; Appendix D.3 provides an overview of baseline methods considered.

D.1 DATASETS

We compared debiasing techniques on several publicly available benchmarking datasets. Most of them are part of IBM's AIF 360 toolkit (Bellamy et al., 2018).[1] In additional experiments described in Appendix E, we considered two synthetic datasets adapted from the previous literature. Below we provide a summary of the datasets. We refer the interested reader to the thorough survey by Quy et al. (2021) for further details.

Adult The Adult Census Income data contains 48,842 instances and includes seven categorical, two binary, and six numerical features. Despite recent scrutiny (Ding et al., 2021), Adult remains a chief tabular benchmarking dataset in the debiasing literature. The task is to predict whether a person's annual income exceeds 50,000$ (Kohavi, 1996; Quy et al., 2021). Potential protected attributes include "*sex*", "*race*", and "*age*".

Bank This dataset was collected during phone call marketing campaigns by a Portuguese banking institution between 2008 and 2013 (Moro et al., 2014; Quy et al., 2021). The dataset comprises 45,211 samples with six categorical, four binary, and seven numerical features. Possible protected variables include "*marital status*" and "*age*". The classification task is to predict a deposit subscription by a potential client.

COMPAS The Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) dataset (Larson et al., 2016; Quy et al., 2021) has generated a lot of discussion within the machine learning community regarding biases and fairness. It was initially publicised as part of ProPublica's analysis.[2] The underlying classification problem is predicting the risk of recidivism. The data comprise 7,214 samples with 31 categorical, 6 binary, and 14 numerical covariates. The protected attribute is "*race*".

MIMIC-III Medical Information Martfor Intensive Care (MIMIC-III-v1.4) database (Johnson et al., 2016) consists of information on the admissions of patients who stayed in critical care units at a large tertiary care hospital. Data includes demographics, vital sign measurements made at the bedside, laboratory test results, procedures, medications, nurse and physician notes, imaging reports, mortality rates, etc. We used preprocessing routine provided by Purushotham et al. (2018) that retains only the first admissions of adult patients ($>$ 15 years). Preprocessed data consist of 17 features from the SAPS-II score. We averaged time-series data for each feature/admission. The underlying classification problem is the prediction of in-hospital mortality. Potential protected variables include "*age*", "*ethnicity*", "*sex*", "*marital status*", and "*insurance type*". In our experiments, we focused on the insurance type and dichotomised this attribute by grouping *Medicare* and *Medicaid* into one category and the rest into another, similarly to Meng et al. (2021).

Synthetic by Loh et al. (2019) Loh et al. (2019) performed extensive simulation experiments comparing subgroup identification methods. Their simulation models are suitable for benchmarking debiasing algorithms. We adopted one of their synthetic datasets with the following generative procedure. For N independent data points:

1. Randomly draw features with marginal distributions given by $X_{1,\,2,\,3,\,7,\,8,\,9,\,10} \sim \mathcal{N}\,(0,\,1)$, $X_4 \sim \mathrm{Exp}(1)$, $X_5 \sim \mathrm{Bernoulli}(\frac{1}{2})$, $X_6 \sim \mathrm{Cat}(10)$ and $\mathrm{corr}\,(X_2, X_3) = 0.5$ and $\mathrm{corr}\,(X_j, X_k) = 0.5$, for $j,\, k \in \{7,\, 8,\, 9,\, 10\}$, $j \neq k$.
2. Randomly draw the protected attribute $A \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$.

[1] https://github.com/Trusted-AI/AIF360

[2] https://github.com/propublica/compas-analysis/

3. Let
$$\text{logit} = \log \frac{\mathbb{P}(Y=1)}{\mathbb{P}(Y=0)} = \frac{1}{2}\left(X_1 + X_2 - X_5\right) + 2\alpha A \mathbf{1}_{\{X_6 \pmod 2 = 1\}}, \tag{9}$$
where $\mathbf{1}_{\{.\}}$ is an indicator function and $\alpha > 0$ is the parameter controlling the magnitude of the correlation between Y and A.
4. Randomly draw the binary classification label $Y \sim \text{Bernoulli}\left(\frac{\exp(\text{logit})}{\exp(\text{logit})+1}\right)$.

Although this dataset is relatively simplistic, the simulation allows controlling the magnitude of classification disparity in the original classifier. In practice, we observe that the higher the value of α, the higher the absolute SPD of an MLP classifier trained on features $X_{1:10}$ and labels Y.

Synthetic by Zafar et al. (2017) Zafar et al. (2017) proposed another simple simulation model for generating datasets with different degrees of disparity in classification outcomes. We extended their model[3] to higher dimensionality and classes that are not linearly separable. The data generating process is specified below. For N independent data points:

1. Randomly draw the binary classification label $Y \sim \text{Bernoulli}\left(\frac{1}{2}\right)$.
2. If $Y = 0$, randomly draw $\tilde{\boldsymbol{X}} \sim \mathcal{N}_2\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}10 & 1\\1 & 3\end{bmatrix}\right)$;
otherwise $\tilde{\boldsymbol{X}} \sim \mathcal{N}_2\left(\begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}5 & 1\\1 & 5\end{bmatrix}\right)$.
3. Let
$$\tilde{\boldsymbol{X}}' = \begin{bmatrix}\cos(\vartheta) & -\sin(\vartheta)\\ \sin(\vartheta) & \cos(\vartheta)\end{bmatrix} \tilde{\boldsymbol{X}}, \tag{10}$$
where ϑ is the rotation angle controlling the correlation between between Y and A.
4. Let
$$\mathbb{P}\left(A=1\right) = \frac{p\left(\tilde{\boldsymbol{X}}' \,|\, Y=1\right)}{p\left(\tilde{\boldsymbol{X}}' \,|\, Y=1\right) + p\left(\tilde{\boldsymbol{X}}' \,|\, Y=0\right)}.$$
5. Randomly draw the protected attribute $A \sim \text{Bernoulli}\left(\mathbb{P}\left(A=1\right)\right)$.
6. Let $g(\tilde{\boldsymbol{x}}) = \boldsymbol{\Theta}_2 \text{ReLU}\left(\boldsymbol{\Theta}_1 \text{ReLU}\left(\boldsymbol{\Theta}_0 \tilde{\boldsymbol{x}} + \boldsymbol{b}_0\right) + \boldsymbol{b}_1\right) + \boldsymbol{b}_2$, where $\boldsymbol{\Theta}_0 \in \mathbb{R}^{h\times 2}$, $\boldsymbol{\Theta}_1 \in \mathbb{R}^{h\times h}$, $\boldsymbol{\Theta}_2 \in \mathbb{R}^{p\times h}$ and $\boldsymbol{b}_0 \in \mathbb{R}^h$, $\boldsymbol{b}_1 \in \mathbb{R}^h$, $\boldsymbol{b}_2 \in \mathbb{R}^p$ are randomly generated matrices and vectors.
7. Let $\boldsymbol{X} = g\left(\tilde{\boldsymbol{X}}\right)$ be a p-dimensional real-valued feature vector.

Similar to the dataset above, this simulation allows controlling the degree of bias by adjusting the parameter ϑ. In practice, values of ϑ closer to zero result in classifiers with higher statistical parity differences.

D.2 Implementation Details

All experiments and methods were implemented in PyTorch (v 1.9.1) (Paszke et al., 2017), based on the code by Savani et al. (2020).[4] We used the same architecture for the classifier $f_{\boldsymbol{\theta}}(\cdot)$ and training scheme for all datasets. We trained a fully connected feedforward neural network with 10 hidden layers, 32 units each, ReLU activations, dropout ($p = 0.05$), and batch normalisation (see Table 2). The network was trained for 1,000 epochs with early stopping by minimising the binary cross-entropy loss using the Adam optimiser (Kingma & Ba, 2015) with `ReduceLROnPlateau` learning rate schedule and mini-batch size of 64.

[3] `https://github.com/mbilalzafar/fair-classification`
[4] `https://github.com/abacusai/intraprocessing_debiasing`

Table 2: Fully connected neural network architecture used in all debiasing experiments. `nn` stands for `torch.nn`; `F` stands for `torch.nn.functional`; `input_dim` corresponds to the number of features d.

	Classifier

1	`nn.Linear(input_dim, 32)`
	`F.relu()`
	`nn.Dropout(0.05)`
	`nn.BatchNorm1d(32)`

2	`for l in range(10):`
	`  nn.Linear(32, 32)`
	`  F.relu()`
	`  nn.Dropout(0.05)`
	`  nn.BatchNorm1d(32)`

3	`out = nn.Linear(32, 1)`
4	`torch.sigmoid()`

Hyperparameters For all datasets, for both pruning and gradient descent/ascent, we set the margin to $\gamma = 0.01$. For bias gradient descent/ascent, we set the learning rate to $\eta = 10^{-5}$, mini-batch size to $M = 256$, and the maximum number of epochs to $E = 100$. For the sake of fair comparison, we used the same margin for the random perturbation baseline (see Appendix D.3).

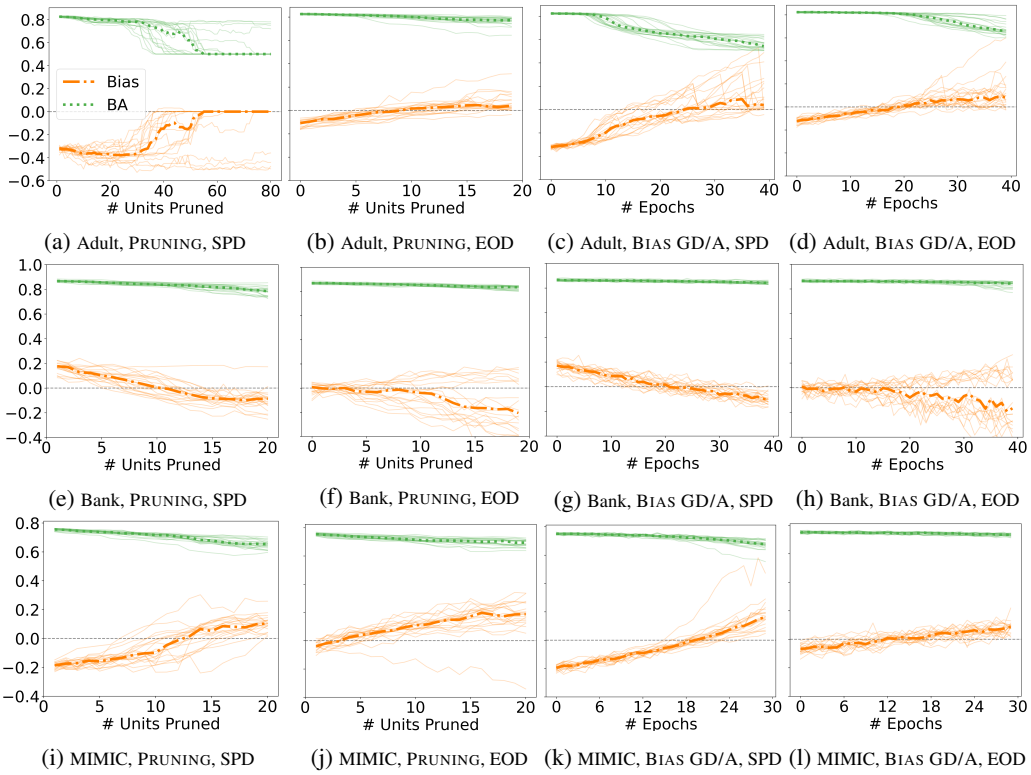
D.3 BASELINES

We compared our pruning and bias gradient descent/ascent procedures to several baseline debiasing methods (see Table 1) summarised briefly below:

- STANDARD refers to the original, potentially biased classifier $f_{\boldsymbol{\theta}}(\cdot)$ with the classification threshold chosen to maximise balanced accuracy on the held-out validation data.
- RANDOM is the random perturbation procedure described by Savani et al. (2020). The weights of the original network $f_{\boldsymbol{\theta}}(\cdot)$ are perturbed by multiplicative Gaussian noise, distributed as $\mathcal{N}(1,\, 0.01)$, independently 500 times. The procedure returns a perturbed network maximising the bias-constrained objective (see Equation 3) on the validation set.
- ROC refers to the reject option classification post-processing algorithm (Kamiran et al., 2012) that swaps classification outcomes for the subjects from the underprivileged group who fall within the confidence band around the decision boundary.
- EQ. ODDS is the equalised odds post-processing method (Hardt et al., 2016). This algorithm adjusts output labels probabilistically to equalise the odds across the protected attribute categories.

E FURTHER RESULTS

E.1 FURTHER QUALITATIVE RESULTS



(a) Adult, PRUNING, SPD (b) Adult, PRUNING, EOD (c) Adult, BIAS GD/A, SPD (d) Adult, BIAS GD/A, EOD

(e) Bank, PRUNING, SPD (f) Bank, PRUNING, EOD (g) Bank, BIAS GD/A, SPD (h) Bank, BIAS GD/A, EOD

(i) MIMIC, PRUNING, SPD (j) MIMIC, PRUNING, EOD (k) MIMIC, BIAS GD/A, SPD (l) MIMIC, BIAS GD/A, EOD

Figure 2: Changes in the **bias**, given by the SPD (a, c, e, g, i, k) and EOD (b, d, f, h, j, l), and **balanced accuracy** of the neural network during pruning (a, b, e, f, i, j) and bias gradient descent/ascent (c, d, g, h, k, l). The results were obtained on Adult (*top*), Bank (*middle*), and MIMIC-III (*bottom*) from 20 train-test splits. **Bold** lines correspond to the median across 20 seeds. Notably, both procedures reduce bias without a considerable effect on accuracy.

E.2 FURTHER QUANTITATIVE RESULTS

Table 3: Balanced accuracy attained after debiasing. STANDARD refers to the original neural network. For each dataset, debiasing was run twice: for the SPD and EOD separately. We report the average accuracy followed by the standard deviation across 20 train-test splits. Best results are shown in **bold**, second-best – in *italic* (except for STANDARD).

Bias Measure	Method	Adult: *Sex*	Bank: *Age*	COMPAS: *Race*	MIMIC-III: *Insurance*
SPD	STANDARD	0.82±0.01	0.86±0.01	0.65±0.01	0.75±0.01
	RANDOM	0.60±0.01	0.60±0.10	0.60±0.03	0.67±0.01
	ROC	**0.79±0.01**	0.66±0.10	0.50±0.00	0.68±0.01
	EQ. ODDS	*0.73±0.02*	0.70±0.02	0.60±0.01	0.57±0.01
	PRUNING	0.56±0.08	*0.84±0.02*	*0.63±0.03*	*0.70±0.02*
	BIAS GD/A	0.66±0.03	**0.86±0.01**	**0.64±0.02**	**0.73±0.01**
EOD	STANDARD	0.82±0.01	0.86±0.01	0.65±0.01	0.75±0.01
	RANDOM	0.78±0.03	**0.86±0.01**	0.61±0.03	**0.75±0.01**
	ROC	**0.82±0.01**	**0.86±0.01**	0.50±0.00	**0.75±0.02**
	EQ. ODDS	0.73±0.02	0.70±0.02	0.60±0.01	*0.57±0.01*
	PRUNING	*0.81±0.01*	*0.85±0.08*	*0.62±0.04*	**0.75±0.01**
	BIAS GD/A	**0.82±0.01**	**0.86±0.01**	**0.63±0.02**	**0.75±0.01**

Table 4: Bias, given by the SPD and EOD, attained after debiasing. STANDARD refers to the original neural network. For each dataset, debiasing was run twice: for the SPD and EOD separately. We report the average bias followed by the standard deviation across 20 train-test splits. Best results are shown in **bold**, second-best – in *italic*.

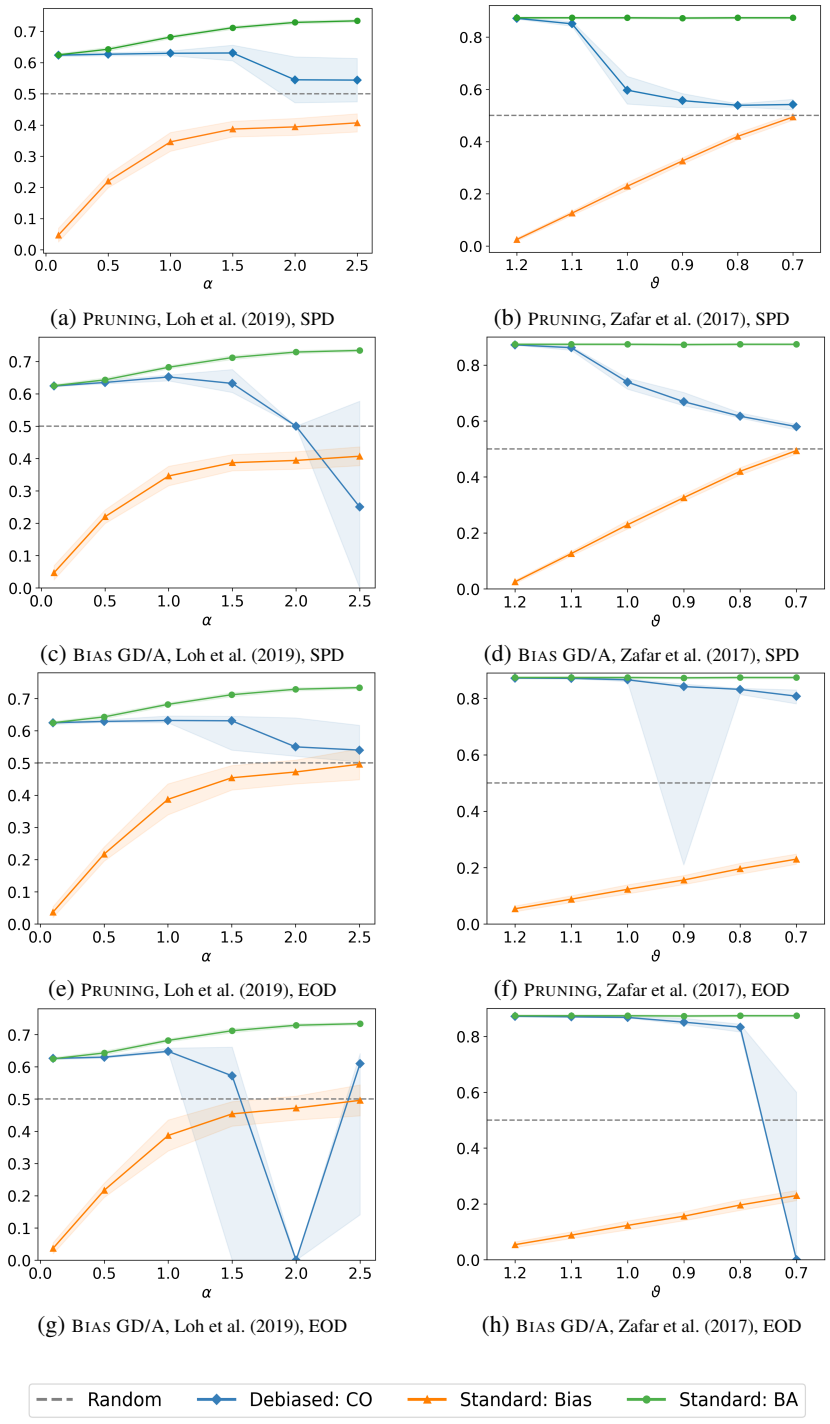
Bias Measure	Method	Adult: *Sex*	Bank: *Age*	COMPAS: *Race*	MIMIC-III: *Insurance*
SPD	STANDARD	-0.32±0.02	0.18±0.04	0.19±0.03	-0.19±0.03
	RANDOM	*-0.04±0.01*	*0.03±0.04*	0.09±0.04	*-0.04±0.01*
	ROC	*-0.04±0.02*	0.08±0.04	**-0.01±0.01**	-0.05±0.01
	EQ. ODDS	-0.09±0.01	0.06±0.03	*0.03±0.06*	**-0.01±0.00**
	PRUNING	*-0.04±0.07*	**0.02±0.02**	*0.03±0.04*	**-0.01±0.03**
	BIAS GD/A	**-0.01±0.04**	*0.03±0.05*	**0.01±0.04**	**-0.01±0.02**
EOD	STANDARD	-0.14±0.02	*0.01±0.04*	0.20±0.05	-0.05±0.04
	RANDOM	-0.07±0.03	0.02±0.04	0.09±0.04	-0.04±0.04
	ROC	-0.05±0.03	0.04±0.04	**-0.01±0.01**	-0.04±0.04
	EQ. ODDS	**-0.01±0.04**	0.04±0.10	0.03±0.06	**0.01±0.04**
	PRUNING	*-0.03±0.03*	*0.01±0.05*	*0.02±0.06*	**-0.01±0.04**
	BIAS GD/A	-0.04±0.03	**0.00±0.06**	*0.02±0.06*	*0.03±0.04*

E.3 SENSITIVITY TO INITIAL CONDITIONS

As observed before, the performance of the debiased classifier can vary considerably, for instance, for the Adult dataset (see Figure 2(a)). To investigate the sensitivity of pruning and bias gradient descent/ascent to initial conditions, in particular, to the bias of the original classifier, we performed further experiments on two synthetic datasets described in Appendix D.1. In both simulations, we trained and debiased neural networks while varying the correlation between the label and protected attribute. We expect debiasing to be less effective when the bias of the original classifier is high.

For the synthetic dataset by Loh et al. (2019), we trained and debiased classifiers under different values of the parameter $\alpha \in [0.0, 2.5]$ (see Equation 9). The resulting SPD varies between approximately 0.0 to 0.4, and the EOD is between 0.0 and 0.5. Figures 3(a) and 3(c) depict changes in the BA and SPD of the original classifier and the bias-constrained objective attained by the model after pruning and bias gradient descent/ascent, respectively. Notably, both methods exhibit similar patterns. For $\alpha \in [0.0, 1.5]$, debiased classifiers retain a BA of approximately 0.60, which corresponds to an unbiased network's performance. In contrast, for $\alpha > 1.5$, the balanced accuracy of the debiased classifier drops closer to 0.5. Bias gradient descent/ascent fails to debias the network for $\alpha = 2.5$. Similar patterns occur when debiasing w.r.t. the EOD (see Figures 2(e) and 2(g))

For the synthetic dataset by Zafar et al. (2017), we varied the value of the parameter $\vartheta \in [0.7, 1.2]$ (see Equation 10). Figures 3(b), 3(d), 3(f), and 3(h) show the results across the range of rotation angles. Analogously to the synthetic dataset by Loh et al. (2019), we observe a drop in the bias-constrained objective of the debiased model for lower values of the parameter ϑ, i.e. under a higher bias. In summary, while proposed techniques successfully mitigate bias, when the bias of the original classifier is relatively high, debiasing leads to a considerable decrease in accuracy.



(a) PRUNING, Loh et al. (2019), SPD

(b) PRUNING, Zafar et al. (2017), SPD

(c) BIAS GD/A, Loh et al. (2019), SPD

(d) BIAS GD/A, Zafar et al. (2017), SPD

(e) PRUNING, Loh et al. (2019), EOD

(f) PRUNING, Zafar et al. (2017), EOD

(g) BIAS GD/A, Loh et al. (2019), EOD

(h) BIAS GD/A, Zafar et al. (2017), EOD

Figure 3: Changes in the **balanced accuracy** and **bias** of the original classifier, given by the SPD (a-d) and EOD (e-h), and the **bias-constrained objective (CO)** (at $\varepsilon \in \{0.05, 0.10\}$, see Equation 3) attained by the network after pruning (a, b, e, f) and bias gradient descent/ascent (c, d, g, h) across varying simulation parameters for the synthetic datasets by Loh et al. (2019) (a, c, e, g) and Zafar et al. (2017) (b, d, f, h). Confidence bounds were constructed across 10 independent simulations, using standard deviations for the SPD and BA and quartiles for the CO.