Deep Learning under Fairness Constraints

Guanqun Yang
Department of Electrical and Computer Engineering
University of California, Los Angeles
Los Angeles, CA 90095
yangguanqun0206@ucla.edu

Lingxiao Wang
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095
lingxw@cs.ucla.edu

Abstract

There is increasing interest in enforcing fairness requirement in algorithmic decision making (ADM) system since biased decision is sometimes still inevitable due to implicit bias embedded in data. We study the problem of fairness constrained deep learning, where the goal is to ensure that sensitive information does not unfairly influence the outcome of a deep neural network. The primal-dual fairness-enforcing algorithm is proposed and its convergence and generalization guarantee is shown. To validate the soundness of our method, we also conduct extensive experiments based on benchmark dataset. Experimental results indicate that our method could effectively reduce bias to as small as 2%.

1 Introduction

Algorithmic decision-making system (ADM) is increasingly widely used in multiple applications including credit-scoring, essay-grading and job applicant selection. However, because of the implicit bias embedded in data that drives such systems, the decisions given by them are often biased against some underrepresented groups and therefore cause discrimination. At the same time, despite the wide applicability of deep neural network in unstructured data like text and image classification, the fairness issues underlying in those applications are yet to be addressed extensively by machine learning community. In this paper, we will study fairness-preserving deep learning with the emphasis of its theoretical guarantee of fairness. We will also evaluate and validate our theoretical using benchmark dataset for fairness machine learning.

Many attempts have been made to formalize and foster fairness in machine learning applications. Dwork et.al first formalized the fairness in classification and this work serves as the prelude of many subsequent works [7]. Chouldechova et.al provided a comprehensive review of various notions of fairness and five promising directions to work on [4]. Most work on fairness-preserving algorithms fall into the three-phase paradigm of machine learning workflow, i.e. preprocessing of data, in-processing of algorithm and postprocessing or predictions. Some notable works include Kamiran et.al.'s preprocessing and postprocessing strategies[9][10] and Kamishima et.al.'s algorithmic modification strategy[11].

2 Methods

We consider the binary classification problem under fairness constraints. More specifically, suppose we have the dataset $S = \{(\mathbf{x}_i, \mathbf{g}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector, $y_i \in \{0, 1\}$ is the label, and $\mathbf{g}_i$ is the protected attribute. For example, $\mathbf{g}_i$ can be race, gender, etc. Our goal is to learn a classifier based on (deep) neural networks under the fairness constraints. In particular, we consider the fairness constraints which can be formulated as the following linear inequality constraint

$$\mathbf{Au}(f) \leq \boldsymbol{c}, \tag{2.1}$$

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

where $\mathbf{A} \in \mathbb{R}^{M \times K}$, $\boldsymbol{c} \in \mathbb{R}^M$ describes the property of the fairness constraints, and $\mathbf{u}(f) \in \mathbb{R}^K$ with each entry $u_i(f)$ denotes a conditional moment that is defined as $u_i(f) = \mathbb{E}\big[h_i\big(\mathbf{X}, \mathbf{y}, \mathbf{G}, f(\mathbf{X})\big) \mid \varepsilon_i\big]$. Note that $h_i : \mathcal{X} \times \mathcal{G} \times \mathcal{Y} \times \text{range}(f) \to [0, 1]$ and ε_i represents an event which depends on $(\mathbf{X}, \mathbf{y}, \mathbf{G})$. It has been shown in [1] that a wide range of definitions of fairness, such as demographic parity and equalized odds, can be transformed into the linear constraint (2.1).

Definition 2.1 (Demographic Parity (DP))**.** A classifier f satisfies demographic parity under a distribution over (X, A, y) if its prediction $f(X)$ is statistically independent of the protected attribute A - that is, if $\mathbb{P}[f(X) = \widehat{y}|A = a] = \mathbb{P}[f(X) = \widehat{y}]$ for all $a, \widehat{y}$. Because $\widehat{y} \in \{0, 1\}$, this is equivalent to $\mathbb{E}[f(X)|A = a] = \mathbb{E}[f(X)]$

Definition 2.2 (Equalized Odds (EO))**.** A classifier f satisfies equalized odds under a distribution over (X, A, y) if its prediction $f(X)$ is conditionally independent of the protecte attribute A given the label y - that is, if $\mathbb{P}[h(X) = \widehat{y}|A = a, Y = y] = \mathbb{P}[h(X) = \widehat{y}|Y = y]$ for all a, y and $\widehat{y}$. Because $\widehat{y} \in \{0, 1\}$, this is equivalent to $\mathbb{E}[f(X)|A = a, Y = y] = \mathbb{E}[f(X)|Y = y]$ for all a, y.

Given the linear inequality fairness constraints in (2.1), we propose to solve the following empirical constrained optimization problem

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell\big(f(\mathbf{x}_i), y_i\big) \quad \text{subject to} \quad \mathbf{A}\widehat{\mathbf{u}}(f) \le \widehat{\boldsymbol{c}}, \tag{2.2}$$

where $\mathcal{F}$ denotes the function class of (deep) neural network, ℓ is the loss function, $\widehat{\mathbf{u}}(f)$ is the empirical conditional moments with each entry $\widehat{\mu}_j = \sum_{i=1}^{n} h_j(\mathbf{x}_i, y_i, g_i, f(\mathbf{x}_i))/n$ given ε_j holds, and $\widehat{\boldsymbol{c}} = \boldsymbol{c} + \boldsymbol{\epsilon}$ with each entry $\epsilon_k \ge 0$, which denotes the relaxation of the fairness constraints in practice. In order to solve the constrained optimization problem (2.2), we follow the method proposed by [1]. More specifically, we propose to find a randomized classifier over the convex hull of the function class $\mathcal{F}$. Intuitively speaking, by adding the fairness constraints, we will reduce the the original function class. Therefore, considering the randomized classifier from the convex hull of the original function class can give us better trade-off between the model accuracy and the fairness constraints. As a result, we propose to solve the following constrained optimization problem

$$\min_{Q \in \Delta} \sum_{f \in \mathcal{F}} Q(f) \frac{1}{n} \sum_{i=1}^{n} \ell\big(f(\mathbf{x}_i), y_i\big) \quad \text{subject to} \quad \mathbf{A} \sum_{f \in \mathcal{F}} Q(f)\widehat{\mathbf{u}}(f) \le \widehat{\boldsymbol{c}}, \tag{2.3}$$

where Q is the randomized classifier, Δ is the set of all distributions over $\mathcal{F}$. The optimization problem in (2.3) can be understood as follows: the convex hull of the function class $\mathcal{F}$ is consists of the finite number of classifiers, and each of them has a probability Q. Our goal is to find the probability Q (the best combination of f's) that minimizes the objective function in (2.3) under the fairness constraints.

For the constrained optimization problem in (2.3), we can get its corresponding Lagrangian as follows

$$L(Q, \boldsymbol{\lambda}) = \sum_{f \in \mathcal{F}} Q(f) \frac{1}{n} \sum_{i=1}^{n} \ell\big(f(\mathbf{x}_i), y_i\big) + \boldsymbol{\lambda}^\top \bigg(\mathbf{A} \sum_{f \in \mathcal{F}} Q(f)\widehat{\mathbf{u}}(f) - \widehat{\boldsymbol{c}}\bigg),$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^K$ is the Lagrangian multiplier. Therefore, the constrained optimization problem in (2.3) is equivalent to the following min-max problem

$$\min_{Q \in \Delta} \max_{\boldsymbol{\lambda} \in \mathbb{R}_+^K} L(Q, \boldsymbol{\lambda}). \tag{2.4}$$

Note that L is convex with respect to Q and $\boldsymbol{\lambda}$, we can solve the this problem by some existing algorithm, such as the exponential gradient algorithm [8]. Since the main focus of this paper is about the generalization performance of the (deep) neural network based classifier as well as how to choose appropriate neural networks to improve the performance of the classifiers, we do not lay out the detailed algorithms for solving this min-max optimization problem.

3 Main Results

In this section, we lay out the main results of this paper.

3.1 Over-parameterized ReLU networks as the function class

In this subsection, we show that why over-parameterized ReLU networks can be a good choice for the fairness constrained binary classification problem. Recall that we are going to solve the min-max problem (2.4), and the key to solve this problem is the following minimization step given the dual variable $\min_{Q\in\Delta} L(Q,\boldsymbol{\lambda})$. Because L is linear in Q, the minimizer can always be chosen to put all of the mass on a single classifier f. Thus, we want to minimize the following loss function with respect to one f

$$L(f,\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(f(\mathbf{x}_i), y_i\big) + \boldsymbol{\lambda}^\top(\mathbf{A}\widehat{\mathbf{u}}(f) - \widehat{\boldsymbol{c}}).$$

Suppose we consider $\ell(f(\mathbf{x}_i), y_i) = \mathbb{1}\{f(\mathbf{x}_i) \neq y_i\}$, we can get

$$\begin{aligned} L(f,\boldsymbol{\lambda}) = -\boldsymbol{\lambda}^\top\widehat{\boldsymbol{c}} + \sum_{m,k}\frac{M_{m,k}\lambda_m}{p_k}\frac{1}{n}\sum_{i=1}^{n} h_i\big(\mathbf{x}_i, y_i, \mathbf{g}_i, f(\mathbf{x}_i)\big)\,\mathbb{1}\{(\mathbf{x}_i, y_i, \mathbf{g}_i)\in\varepsilon_k\} \\ + \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{f(\mathbf{x}_i)\neq y_i\}, \end{aligned}$$

where $p_j = \widehat{\mathbb{P}}(\varepsilon_j)$. Let us define C_i^0 and C_i^1 as follows

$$\begin{aligned} C_i^0 &= \mathbb{1}\{y_i \neq 0\} + + \sum_{m,k}\frac{M_{m,k}\lambda_m}{p_k}\frac{1}{n}\sum_{i=1}^{n} h_i\big(\mathbf{x}_i, y_i, \mathbf{g}_i, 0\big)\,\mathbb{1}\{(\mathbf{x}_i, y_i, \mathbf{g}_i)\in\varepsilon_k\}, \\ C_i^1 &= \mathbb{1}\{y_i \neq 0\} + + \sum_{m,k}\frac{M_{m,k}\lambda_m}{p_k}\frac{1}{n}\sum_{i=1}^{n} h_i\big(\mathbf{x}_i, y_i, \mathbf{g}_i, 1\big)\,\mathbb{1}\{(\mathbf{x}_i, y_i, \mathbf{g}_i)\in\varepsilon_k\}, \end{aligned}$$

we can obtain that minimize $L(f,\boldsymbol{\lambda})$ is equivalent to minimize the following cost-sensitive objective function

$$\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)C_i^1 + (1 - f(\mathbf{x}_i))C_i^0. \tag{3.1}$$

This formulation tells us that we can actually solve the minimization problem $\min_{Q\in\Delta} L(Q,\boldsymbol{\lambda})$ even the constraint depends on f. Furthermore, by plugging the definition of C_i^0 and C_i^1, we can see that the minimizer of the cost-sensitive problem in (3.1) is the one that can exactly correctly classify all the data. Inspired by this observation, we propose to use the over-parameterized ReLU networks as the function class for the fairness constrained binary classification problem since it can achieve zero training loss [13, 6, 2] as well as main good generalization performance [3] by using the gradient based training algorithms. As a result, we propose to solve the inner minimization problem by replace the f in (3.1) with the loss of training neural networks.

3.2 Generalization performance

In this subsection, we lay out a preliminary generalization results of our proposed method using L-layer neural networks as our function class. More specifically, we consider the following function class

$$\mathcal{F} = \{f : \mathbb{R}^d \to \mathbb{R} : \|\mathbf{W}_i\|_2 \leq s_i, i = 1,\ldots,L\},$$

where we have f to be a neural network of the following form

$$f(\mathbf{x}) = \mathbf{v}^\top\sigma_L(\mathbf{W}_L\sigma_{L-1}(\mathbf{W}_{L-1}\ldots\sigma_1(\mathbf{W}_1\mathbf{x})\ldots)), \tag{3.2}$$

where $\sigma_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$ is the ReLU activation function, $\mathbf{W}_i \in \mathbb{R}^{d_i\times d_{i-1}}$ is the weight matrix for $i = 1,\ldots,L$, $\mathbf{v}\in\mathbb{R}^{d_L}$. In addition, we assume that $\|\mathbf{v}\|_2 \leq V_d$ and $\max_i \|\mathbf{x}_i\|_2 \leq B$. We have the following results.

Theorem 3.1. Let $(\widehat{Q}, \widehat{\boldsymbol{\lambda}})$ be the solution of the min-max problem in (2.4). There exists some constants $\{c_i\}_{i=1}^3$ such that if we set the relaxation ϵ_m as $\epsilon_m \geq c_1 \sum_k |M_{m,k}|(\sqrt{\log(1/\delta)/n_k})$, where $n_k = |\{i : (\mathbf{x}_i, \mathbf{g}_i, y_i) \in \varepsilon_m\}|$, then with probability at least $1 - c_2\delta$, we have

$$\mathbb{E}[\ell(\widehat{Q}(X), Y)] - \mathbb{E}[\ell(Q^*(X), Y)] \leq c_3 \frac{\log(1/\delta)}{\sqrt{n}} + c_4 \frac{B\sqrt{LV_d}\Pi_{i=1}^L s_i}{\sqrt{n}},$$

where Q^* is the minimizer of the population loss. In addition, we have the output $\widehat{Q}$ satisfies all the fairness constraints.

Remark 3.2. According to Theorem (3.1), we can see that the minimizer of the min-max problem satisfies all the fairness constraint. In addition, it will have the excess risk at the order of $O(n^{-1/2})$. Note that in practice, we cannot exactly solve the min-max problem. Thus, there will be an extra optimization term in our final results when we take into account the effect of different optimization algorithms.

Remark 3.3. In the current paper, we haven't figure out the effect of the fairness constraints on the Rademacher complexity of the neural network function class we considered. Therefore, we follow the proof procedure in [1] to establish the excess risk of our method. To prove this result, we only need to get the Rademacher complexity of the function class we considered. According to the proof of Theorem 1 in [12], we get the Rademacher complexity of the function class we considered at the order $\widetilde{O}(B\sqrt{LV_d}\pi_{i=1}^L s_i/\sqrt{n})$. The remaining step is to make use of the same proof stratage as in the proof of Theorem 4 in [1]. Since this part is not our contribution, we just ignore the detailed proofs here.

4 Experiment

In order to validate the soundness of our theoretical analysis, we choose the relatively simple two-layer parameterized ReLU neural network as our model. Additionally, adult performance dataset [5] is chosen for comparison purposes with other literature [1].

The evaluation for fairness preserving algorithms generally involve error and violation of fairness metrics, which are defined as

$$\text{Error} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{\widehat{f}(\mathbf{x}_i) \neq y_i\right\} \quad \text{Violation} = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{f}(\mathbf{x}_i) - \widehat{f}(\mathbf{x}_i|a_i)\right|$$

where we note that error is empirical 0-1 loss while fairness violation is the empirical absolute difference between prediction and conditional prediction.

At the same time, when the choice of tolerance for fairness violation ϵ varies, tradeoff between error and fairness might emerge. Specifically, we will consider

- Evaluating error and fairness violation as a function of training iteration.
- Evaluating tradeoff between error and fairness violation as fairness violation ϵ changes.

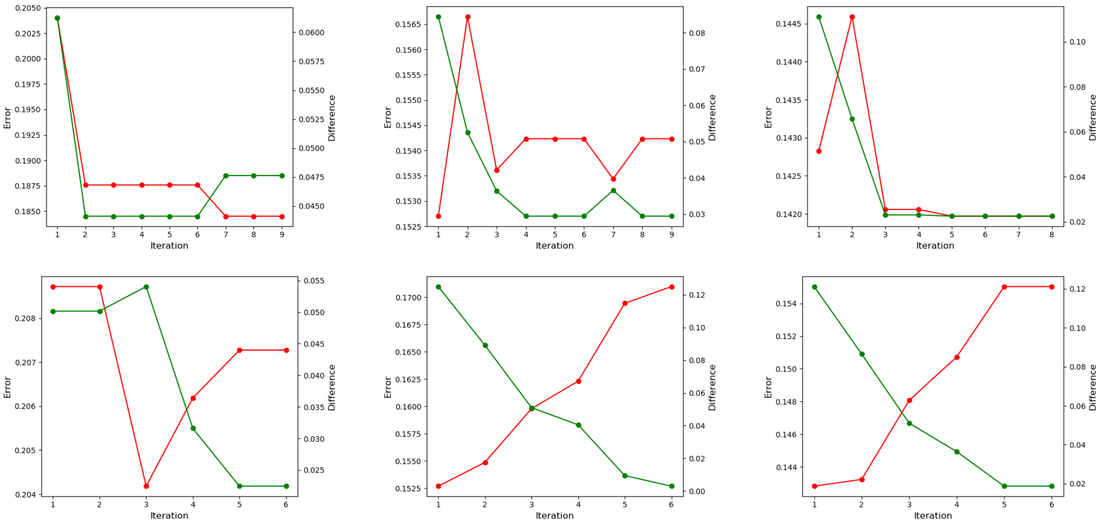
In the following two subsections, we will address both of them and provide comparative analysis with other machine learning algorithms including Logistic regression and AdaBoost.

4.1 Convergence Visualization

As is shown in Figure 1, our algorithm could converge within 10 iterations and in some cases (optimizing over EO in Logistic regression), the convergence is made possible with only 6 iterations. Note that our algorithm terminates when duality gap becomes smaller than predefined threhold ν and therefore no consistent decrease in both error and fairness violation is guaranteed.

When comparing performance of different fairness metrics, it could be seen that the optimization over EO could provide more desirable results for *all* three models, where *both* fairness violation and error are low when iteration terminates. This probably result from its relatively simpler conditional expectation formulation than DP since EO does not require conditioning on protected attributes $\mathbf{a}_i$.

Across three different models, it is worthwhile to note that neural network does *not* necessarily provide better results than the other two. One explanation for this is that dataset we use is *structured* while neural network models are more applicable for applications involving *unstructured* data like image, audio and text. Similar experimentation procedures could be carried out when such datasets are available.



Neural net Logistic regression AdaBoost

Figure 1: Error (red) and fairness violation (green) with respect to number of iterations under EO (up) and DP (down)

4.2 Tradeoff Visualization

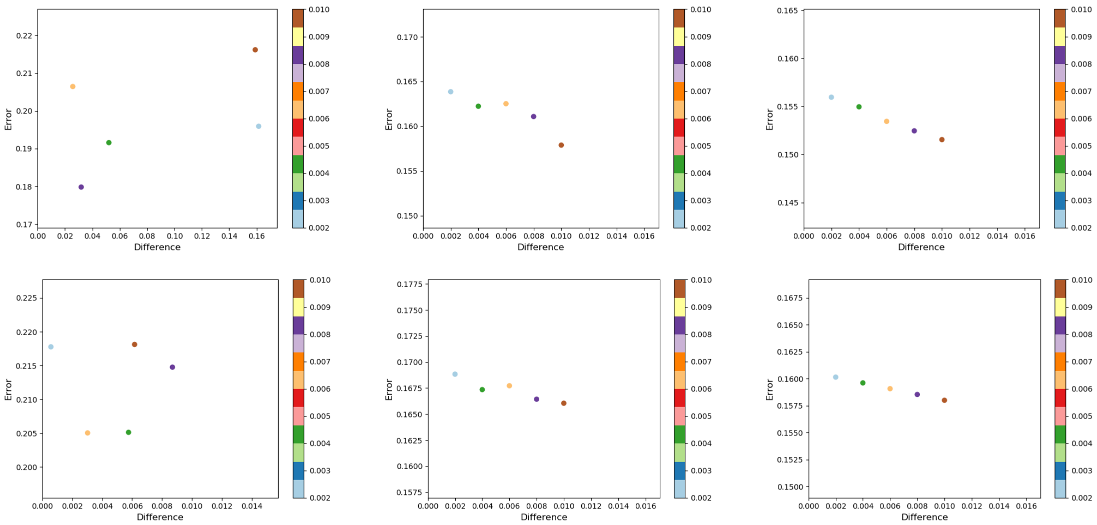
When the fairness violation parameter ϵ is set to $0.002, 0.004, \cdots, 0.01$, the resulting error and fairness violation could be computed with testing data and this tradeoff shown in Figure 2.

Since fairness parameter ϵ mandates the level of fairness violation user could accept, it is generally expected to see the increase in fairness violation when ϵ is set larger. However, despite expected behavior of both Logistic regression and AdaBoost, it is not the case for neural network, where inconsistent violation-error tradeoff appears. Specifically, when optimizing over EO, the largest tolerance ($\epsilon = 0.1$) and smallest tolerance ($\epsilon = 0.02$) give almost the same fairness violation albeit the latter's significant reduction in error. At the same time, what is also counter-intuitive is that best result is yielded when $\epsilon = 0.08$, in which both fairness violation and error is the lowest among five different choices of ϵ. Similar phenomenon also emerges when optimizing over DP, where we observe that better fairness guarantee might *not* increase error. This inconsistent behavior makes the tradeoff evident for other two models hardly visible in neural network.

One way to interpret this behavior is the non-convexity of neural network. Both Logistic regression and AdaBoost could be formulated as a convex optimization problem, the global optimal could be attained using our algorithm. However, it is not likely for neural network to attain global optimal with our algorithm. Therefore, every point shown in the figure correspond to local optimums, where there are no guarantees for consistent behavior when trading off fairness violation and error.

Therefore, we note that when adopting neural networks to practical fairness preserving machine learning system, fairness violation tolerance ϵ would be an additional hyperparameter to tune to achieve best performance.

At the end of this section, we present a preliminary result using over-parameterized neural networks. It can be seen from Figure 3 that, the results of over-parameterized neural networks are better than shallow networks. Due to the time limit, we do not fine tune our results with respect to the over-parameterized neural networks. We believe we can get much better results in our future work.



Neural net Logistic regression AdaBoost

Figure 2: Testing tradeoff with different ϵ under metric EO (up) and DP (down)

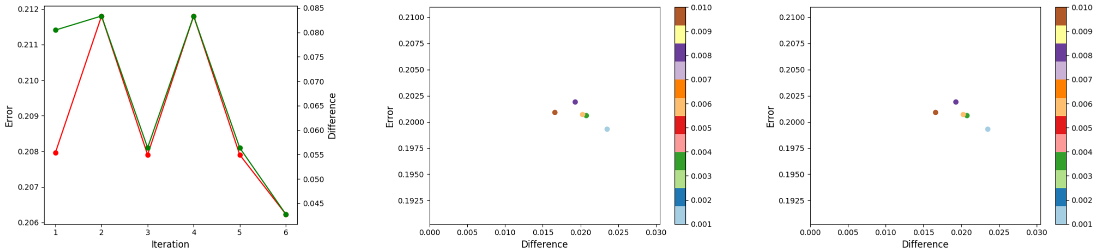


Figure 3: Results of ver-parameterized networks under metric EO

5 Conclusion and Future Work

In this work, we investigate the performance of deep learning for the classification problem under the fairness constraints. More specifically, we propose to use over-parameterized neural networks for fairness constrained classification problem. In addition, we provide a generalization performance for a specific class of neural networks. For our future work, we will fully study the empirical performance of the over-parameterized neural networks for this problem. In addition, we will prove tighter generalization performance of our method by study the effect of the fairness constraints of the neural network function class.

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*, 2018.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

[3] Yuan Cao and Quanquan Gu. A generalization theory of gradient descent for learning over-parameterized deep relu networks. *arXiv preprint arXiv:1902.01384*, 2019.

[4] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

[5] Paulo Cortez and Alice Maria Gonçalves Silva. Using data mining to predict secondary school student performance. 2008.

[6] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[7] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

[8] Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *COLT*, volume 96, pages 325–332. Citeseer, 1996.

[9] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

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

[11] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 643–650. IEEE, 2011.

[12] Xingguo Li, Junwei Lu, Zhaoran Wang, Jarvis Haupt, and Tuo Zhao. On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond. *arXiv preprint arXiv:1806.05159*, 2018.

[13] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.