Analysis of the paper:

Fairness via Representation Neutralization

By

2019A7PS0061G Anand S

2019A7PS0026G Mrudul M Nair

2019A7PS0015G Shubham Kumar

Prepared in Partial Fulfillment of the course

CS F425 Deep Learning

Birla Institute of Technology and Science (BITS), Pilani

May 2022

Table of Contents

1. Introduction

Fairness- Why we care and Definition

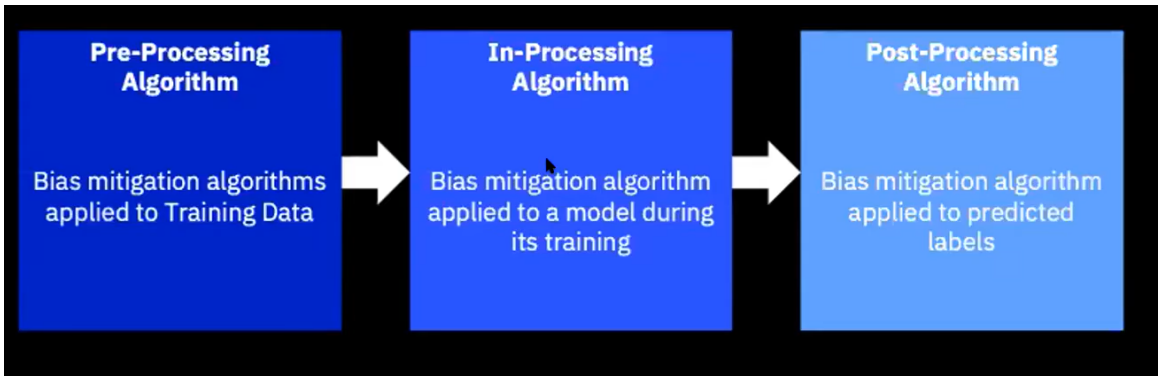
AI can embed our own human biases and deploy them at scale. A classic example of this is COMPAS which incorrectly labeled African-American defendants as high risk at twice the rate of white defendants.
Many experts believe bias may be the major barrier that prevents AI from reaching its full potential.

How do we measure Fairness?

The method we chose is the one we felt was the most intuitive-Demographic Parity
Demographic Parity states that the proportion of each segment of a protected class (e.g. gender) should receive the positive outcome at equal rates.

When could Bias be removed?

Any stage in the pipeline! That is our main motivation for the project--
Try removing bias in different stages of the pipeline and find the best practices for our particular dataset with the hope it could be generalized.

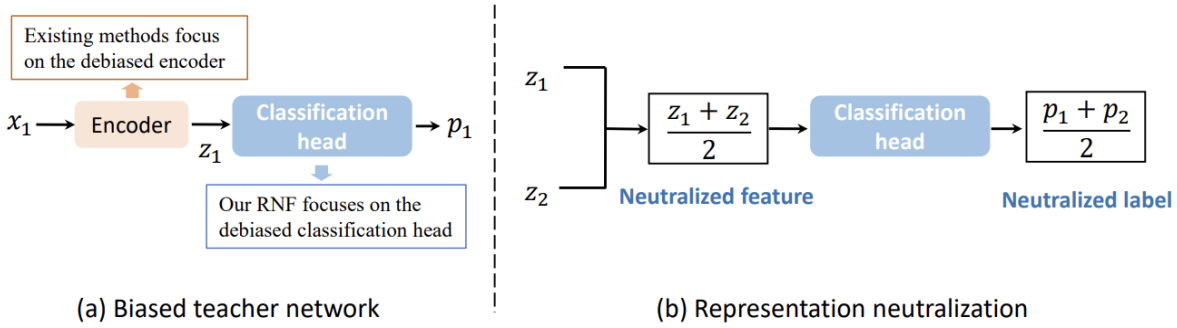


In our analysis we will be focusing on the In-Processing stage, since for deep learning there are not many good methods for bias mitigation at the In-Processing stage, and we explore a novel method for the same called Fairness via Representation Neutralization.

Fairness via Representation Neutralization(The Paper)

This paper explores the question “Can we reduce the discrimination of DNN models by only debiasing the classification head, even with biased representations as inputs?”. Debiasing only the classification head is less complex than existing bias mitigation methods that primarily focus on learning debiased encoders which requires a lot of instance-level annotations for sensitive attributes and also does not guarantee that all fairness sensitive information has been removed from the encoder. This is achieved by using the technique - Representation Neutralization for Fairness (RNF), which is a 2 step process. In the first step, we train the model using cross entropy loss, and obtain a biased teacher network. During the second step, we freeze only the encoder part and use it as our backbone encoder for learning representations. We then re-train only the classification head using feature neutralization by using samples with the same ground-truth label but different sensitive attributes. The key idea is to discourage the classification head from capturing undesirable correlation between fairness sensitive information in encoder representations with specific class labels and thereby reduce discrimination of DNN models with minimal degradation in task-specific performance.

We implemented this paper as part of our midsem submission. After that we were able to improve the accuracy by 7 percent by just normalizing the features before passing to the model.



(a) Biased teacher network (b) Representation neutralization

Dataset

The dataset we are using is Law School Admission Data.
It was collected by Law School Admission Council, and involved a survey of 21,790 students from 163 law schools.
We aim to predict the first year average grade (FYA) of a student from the dataset. The dataset includes several features including their entrance exam scores (LSAT), grade-point average (GPA) collected prior to law school, race, sex etc.

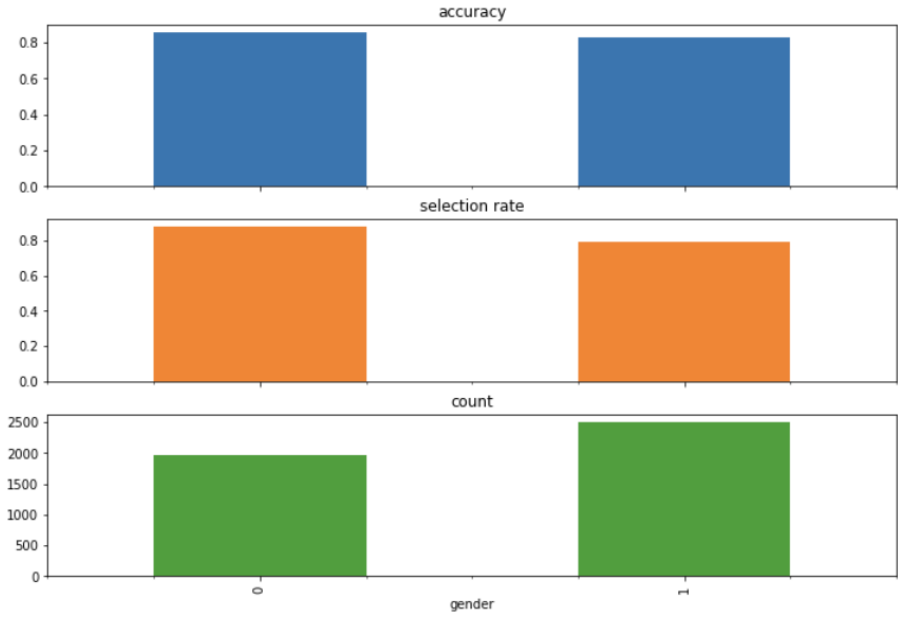
2.Analysis

[Note: For the analysis below, Gender 0 corresponds to Female, 1 corresponds to male]
We had 2 goals for our analysis-
1.Compare RNF to other Fairness Methods
2.See how RNF varies based on alpha(main hyperparameter) and study bias-error tradeoff using Pareto Curves

Basically we want to see if our goal is to get a great fairness,accuracy combo, should we go for Deep Learning since even though it can provide great accuracy, there are not many DL specific methods for fairness.

Part 1: Comparing RNF

For RNF



gender	accuracy	selection rate	count
0	0.860182	0.878419	1974
1	0.82848	0.795772	2507

Accuracy=0.842
Demographic Parity Difference=0.08

Contender 1: DL Model without RNF

Here we use the exact same DL model used above, but remove the representation neutralization part.

gender	accuracy	selection rate	count
0	0.927951	0.753194	1957
1	0.932647	0.650555	2524

Accuracy=0.93
Demographic Parity Difference=0.10

We can see accuracy increases but bias decreases as expected. However the tradeoff seems worth it as accuracy gets a strong bump.

Contender 2: ML Model without Bias Mitigation

gender	accuracy	selection_rate	count
0	0.807444	0.879729	3251
1	0.786335	0.710526	4142

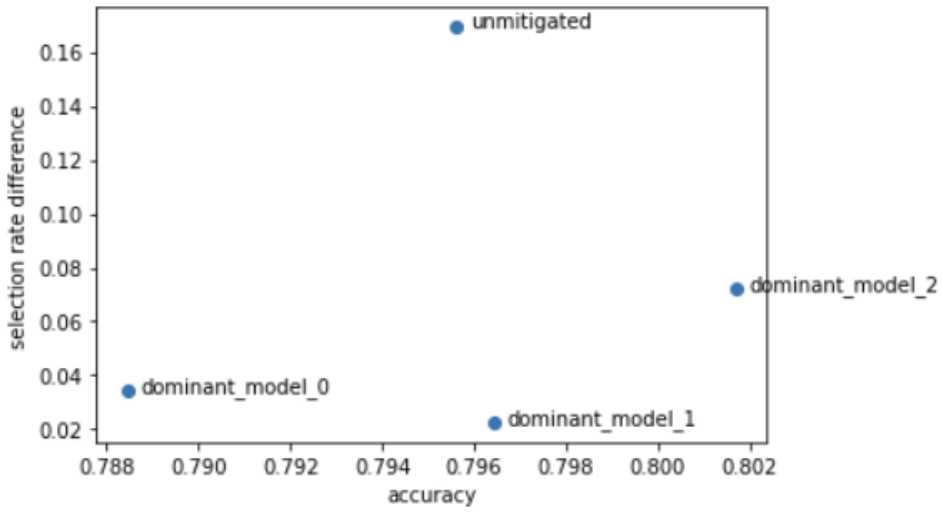
Accuracy=0.795
Demographic Parity Difference=0.169

Here we use a simple logistic regression and as expected this performs worse than previous 2 models both in terms of accuracy and fairness.

Contender 3:ML Model with Bias Mitigation

For the final battle, to use ML or DL, we use how the previous ML model's bias changes when we use an in-processing bias mitigation algorithm called grid search reduction.

Grid search reduces fair classification to a sequence of cost-sensitive classification problems, returning the deterministic classifier with the lowest empirical error subject to fair classification constraints among the candidates searched



(here unmitigated corresponds to the logistic regression used above)
Demographic parity difference for-
Dominant_model_0=0.034
Dominant_model_1=0.022
Dominant_model_2=0.0722

ML with bias mitigation performs much better(dominant models 0 and 1) than RNF in terms of reducing bias, however they have poorer accuracy.

Part 2: Pushing RNF to its Limits

We faced a lot of difficulty while trying to implement the paper as the model wasn't learning anything for our dataset. We contacted the original authors and they said the hyperparameter(alpha) we use GREATLY(our model was not able to learn anything when we used the alpha recommended by the authors for the dataset used in the paper) affects the performance. In this section we will be exploring alpha(it is a part of the loss function).

Loss function

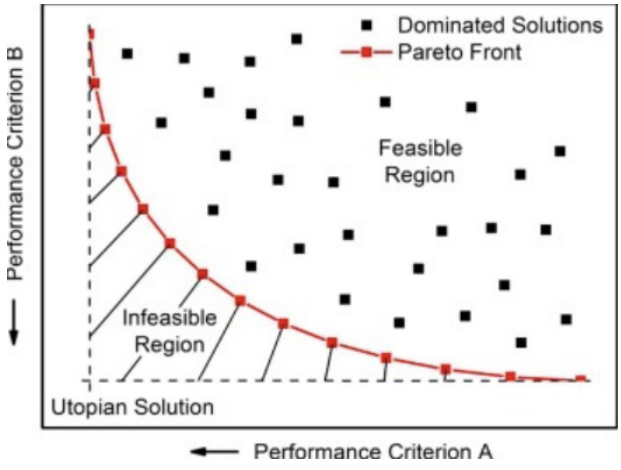
$$\mathcal{L}_{\text{Smooth}} = \sum_{\lambda \in [\frac{1}{2}, 1)} |c(\lambda z_1 + (1-\lambda) z_2) - c(\frac{1}{2} z_1 + \frac{1}{2} z_2)|_1. \quad (3)$$

$$\mathcal{L}_{\text{MSE}} = (\hat{y}_i - y)^2 = \{c(\frac{1}{2} z_1 + \frac{1}{2} z_2) - (\frac{1}{2} p_1 + \frac{1}{2} p_2)\}^2. \quad (1)$$

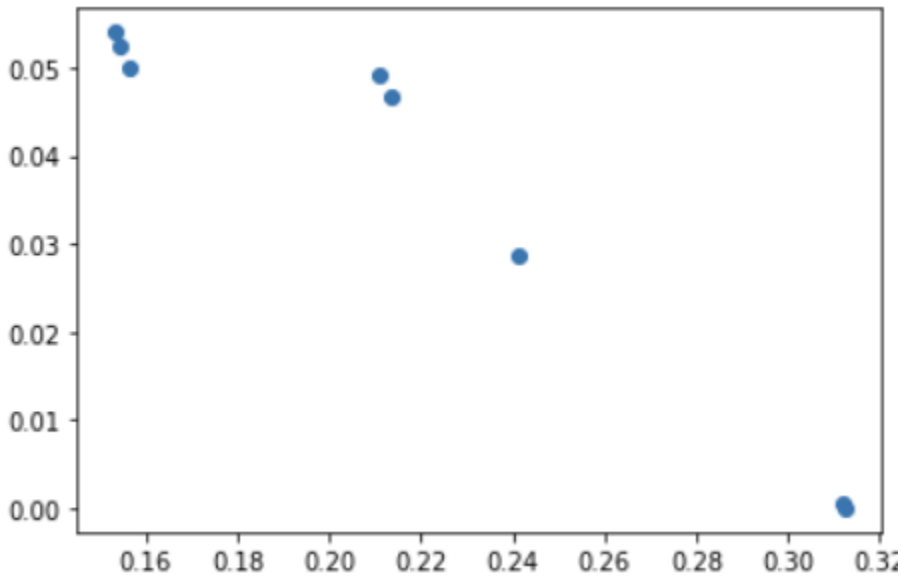
$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \alpha \mathcal{L}_{\text{Smooth}}. \quad (4)$$

Since we are trying to minimize 2 values here(fairness and bias), we will be plotting pareto fronts. It allows the designer to restrict attention to a set of efficient choices, rather than considering the full range.

A generic pareto front--



Because of time constraints, we used only 20 points for pareto(of which 7 made it to pareto front). Further, since RNF takes a lot of time to train we reduced the epochs from 70(50+20) to 13(9+4).



In the graph above, error is plotted on x axis, and bias(demographic parity difference) on y axis.

The advantage of plotting such a curve is now we have flexibility to choose a model with extremely low error, but bias around 0.05, or a model with extremely low bias, but error around 0.325. However it might be wiser to choose a point in the middle(0.25 error, and 0.03 bias) with decent accuracy and decent fairness

The values of alpha we chose were quite different from what the authors used for their dataset. Primarily it was by trial and error and we found our dataset required extremely small values of alpha. So the values of alpha we used were in the range [0,0.0002].

3. Findings and Contribution

RNF produced great results for our dataset. DL without RNF produced better accuracy and ML with bias mitigation gave less bias, however no model we tested gave both a better accuracy and better bias mitigation than RNF.

The major problems we observed were-

1. Difficulty in implementation.
2. Significantly longer time to train because of the representation neutralization step.
3. Performance is extremely dependent on the value of alpha, and since it takes a lot of time to train, it is extremely time consuming to find the right alpha, and we don't even know IF there is a right alpha. In Fact, for most of the alpha values we tested, the model didn't even work.
4. Small decrease in bias compared to the big drop in accuracy for our dataset

However authors showed RNF works really well in other datasets(more complex ones), so if one ends up using this, main recommendation would be to plot pareto fronts as the model is extremely sensitive to alpha values, so the graph would greatly assist to visualize the behavior.