Flexible Regularization Approaches for Fairness in Deep Learning

Matt Olfat[1], Yonatan Mintz[2]
[1]UC Berkeley, IEOR
[2]Georgia Tech, ISyE

Abstract

Artificial Neural Networks (ANN) have been shown to be uniquely effective for many predictive tasks, such as image recognition and natural language processing. However, as they have become more ubiquitous, there have been several examples of these models exhibiting anthropomorphic bias (e.g. making predictions correlated with race or gender for unrelated tasks) due to over fitting, amplifying and systematizing bias already inherent in training data. To address this problem, we consider a novel regularization approach for deep learning, inspired by the constrained optimization literature, that directly penalizes unwanted disparities in treatment of populations proportionally to their impact on observed bias. Using this method, we can control bias at training time, as opposed to in a pre- or post-processing step; this results in concurrent out-of-sample improvements in both fairness and accuracy for some data sets. Our methods fit well into existing optimization and training approaches and can be easily generalized across network architectures and notions of fairness. We validate our methods empirically on several real world data sets that contain implicit bias. Namely we consider the impact of race on recidivism prediction, gender on income, and wine color on quality.

Notions of Fairness

$z_i \in \{-1,+1\}$: protected label

$x \in \mathcal{X}$: data feature tuple

$y \in \{-1,+1\}$: target label

$\varphi \in \mathcal{F}$: an ANN trained for the classification task of interest

$\sigma : \mathcal{X} \times \Theta \to [0,1]$: final activation layer of φ

- Referred to as "disparate impact" in the literature
  - Decision uncorrelated to protected class

$$\left| \underbrace{\mathbb{P}\left[\varphi(x_i) = +1 \middle| z = +1\right]}_{\text{True-positive rate}} - \underbrace{\mathbb{P}\left[\varphi(x_i) = +1 \middle| z = -1\right]}_{\text{False-positive rate}} \right| \leq \Delta$$

- Alternatively, equalized opportunity (Hardt et al., 2016)
  - Require no correlation when restricted to those "deserving"
  - Can also be implemented within our framework

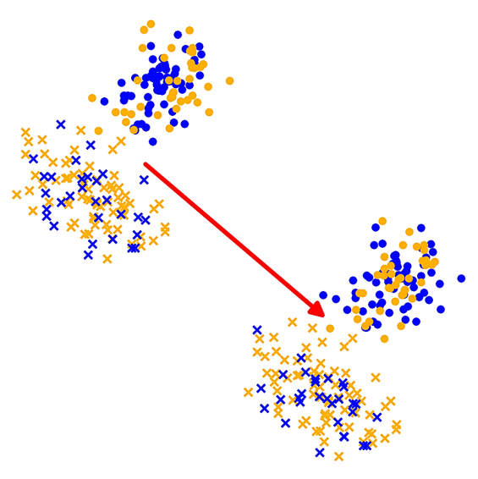
$$\left| \mathbb{P}\left[\varphi(x_i) = +1 \middle| z = +1, y = +1\right] - \mathbb{P}\left[\varphi(x_i) = +1 \middle| z = -1, y = +1\right] \right| \leq \Delta$$

- Our approach to approximately solve the constrained training problem:

$$\min_{\varphi \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(\varphi(x_i), y_i)$$
$$s.t.\ G(\varphi, \{x_i\}_{i=1}^n, \{y_i\}_{i=1}^n, \{z_i\}_{i=1}^n) \leq \Delta$$

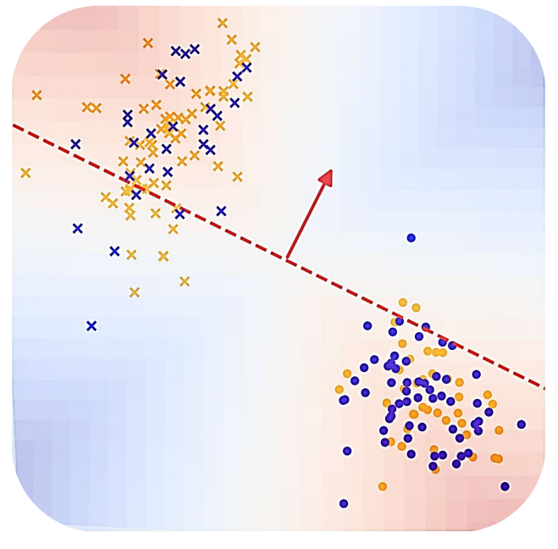
- Where G is one of the fairness notions stated above

Related Approaches



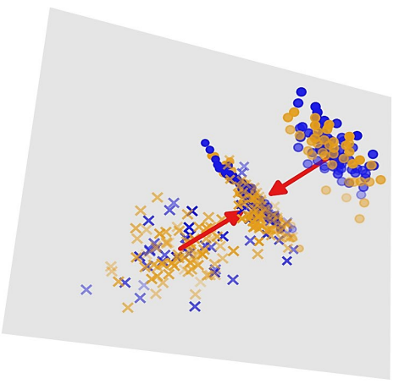
Post-Processing

- Pedreschi et al (2008)
- Kamiran & Calders (2009)
- Luong et al (2011)
- Hardt et al (2016)



In training

- Calders & Verwer (2010)
- Kamishima et al (2011)
- Zliobaite et al (2015)
- Zafar et al (2017)
- Agrawal et. Al (2018)
- Olfat & Aswani (2018)



Pre-Processing

- Calders et al (2011)
- Dwork et al (2011,2012)
- Zemel et al (2014)
- Feldmen et al (2015)
- Olfat & Aswani (2018)

These methods were mostly developed for convex training problem with small data, they do not generalize well to deep learning

Adversarial Deep Learning Approaches:

- Edwards and Storkey (2015)
- Beutel et al (2017)
- Madras et al (2018)
- Zhang et al (2018)

Pre-Processing and In Training :

- Bolukbasi (2016)
- Burns et al (2018)

These methods often involve either large parameter counts or are highly model specific, our method is more general and requires fewer parameters.

Fairness Constraints and Regularization

- Consider the optimization problem:

$$\min_{\varphi \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(\varphi(x_i), y_i)$$
$$s.t.\ G(\varphi, \{x_i\}_{i=1}^n, \{y_i\}_{i=1}^n, \{z_i\}_{i=1}^n) \leq \Delta$$

- Need to reformulate to fit SGD
- General method: relax constraint to form regularization
- Lagrangian relaxation:

$$\min_{\varphi \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(\varphi(x_i), y_i) + \lambda(G(\varphi, \{x_i\}_{i=1}^n, \{y_i\}_{i=1}^n, \{z_i\}_{i=1}^n) - \Delta)$$

- Simlar to l-2 and l-1 regularization that can be viewed as Lagrangian relaxations

Higher Order Interaction Terms

Issue

- Objective not compatible with SGD methods
  - The term $G(\varphi, \{x_i\}_{i=1}^n, \{y_i\}_{i=1}^n, \{z_i\}_{i=1}^n)$ confounds data in objective (and gradient)
  - Need additive decomposition by data

Resolution

- Use higher order moment constraints
  - $|\mathbb{E}[\sigma(x_i)^m z_i]| \leq \delta_m.$
  - δ different for each power
  - Similar to constraints in Zafar et al. (2017)
  - Effectively a moment matching constraints on X_+ and X_-

Proposition 1. *For appropriate values of $\lambda, \mu \geq 0$, the objective function for the relaxed higher order interaction constraints can be expressed as:*

$$\sum_{i=1}^{n} \mathcal{L}(\varphi(x_i), y_i) + (\lambda - \mu)(\frac{1}{n}\sum_{i=1}^{n} \sigma(x_i)^m z_i),$$

and the corresponding gradient signal is given by:

$$\sum_{i=1}^{n} \nabla_\theta \mathcal{L}(\varphi(x_i), y_i) + (\lambda - \mu)(\frac{m}{n}\sum_{i=1}^{n} \sigma(x_i)^{m-1} z_i \nabla_\theta \sigma(x_i)).$$

Remark 1. *For the case of the first order term (i.e. $m = 1$) and binary protected class, the resulting regularizer is equivalent to the demographic parity regularizer, but with the activation function as opposed to the prediction label.*

Remark 2. *For the case of the first order term (i.e. $m = 1$) and binary protected class, the resulting regularizer can be modified to encode equal opportunity by performing a similar procedure on the constraint:*

$$\left|\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[y_i = 1]\sigma(x_i)^m z_i\right| \leq \delta_m.$$

Empirical Results

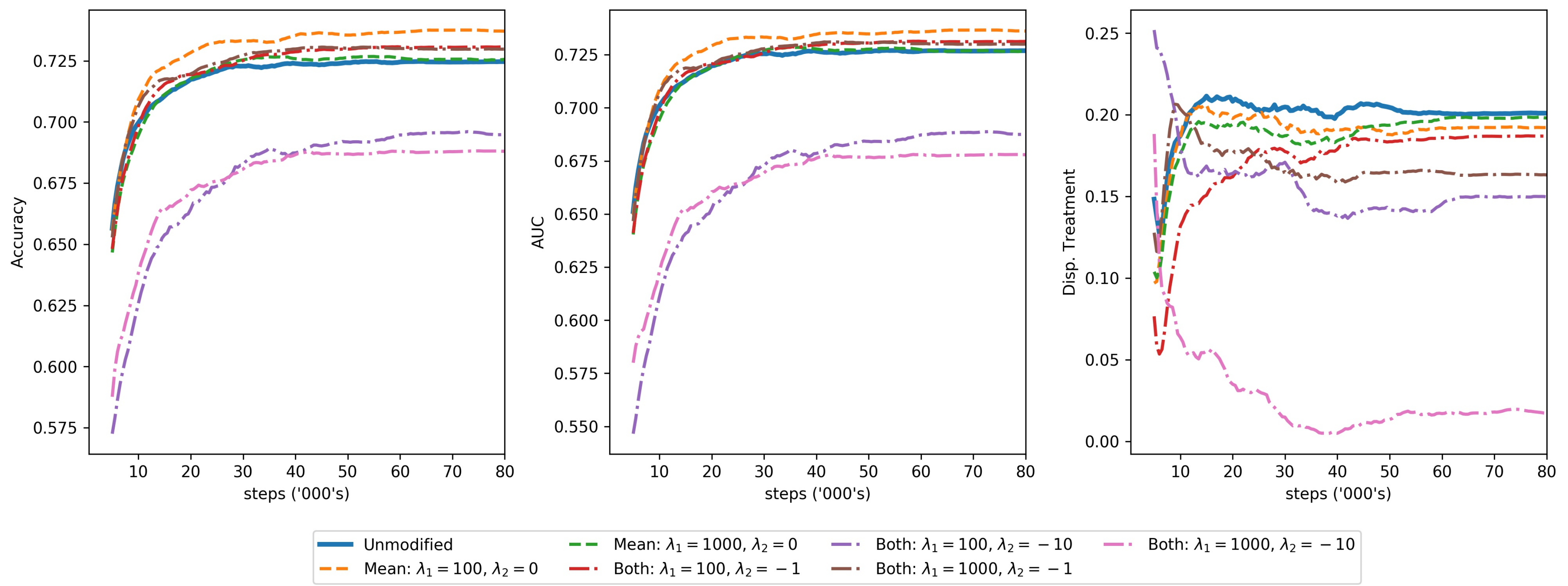


Table 1: Results for the Adult Income dataset on a single-layer network architecture

VERSION	METRIC PARAMS	ACCURACY	AUC	DP
UNMOD.	0, 0	0.7883	0.6238	0.0489
PRE	0, 0	0.7149	0.6186	0.3173
POST	0, 0	0.7967	—	0.0465
FIRST	-100, 0	0.7470	0.6226	0.0498
	-1000, 0	0.7799	0.6189	0.0459
SECOND	-100, 1	0.7894	0.5893	0.0337
	-100, 10	0.7824	0.5818	0.0283
	-1000, 1	0.7942	0.5925	0.0336
	-1000, 10	0.7819	0.5814	0.0292

Table 2: Results for the Wine Quality dataset on a single-layer network architecture

VERSION	METRIC PARAMS	ACCURACY	AUC	DP
UNMOD.	0, 0	0.7246	0.7268	0.2009
PRE.	0, 0	0.6333	0.6231	0.1925
POST.	0, 0	0.6877	—	0.1654
FIRST	100, 0	0.7372	0.7362	0.1925
	1000, 0	0.7259	0.7272	0.1979
SECOND	100, -1	0.7307	0.7313	0.1870
	100, -10	0.6957	0.6884	0.1473
	1000, -1	0.7296	0.7298	0.1617
	1000, -10	0.6880	0.6780	0.0173

Table 3: Results for the Recidivism dataset on a single-layer network architecture

VERSION	METRIC PARAMS	ACCURACY	AUC	EO
UNMOD.	0, 0	0.6761	0.6758	0.3073
PRE	0, 0	0.5476	0.5486	0.0471
POST	0, 0	0.5057	—	0.0001
FIRST	100, 0	0.6705	0.6706	0.2994
	1000, 0	0.6761	0.6756	0.3056
SECOND	100, -1	0.6676	0.6678	0.2798
	100, -10	0.6098	0.6126	0.1747
	1000, -1	0.6723	0.6722	0.3001
	1000, -10	0.6089	0.6122	0.1071

Table 4: Results for the Adult Income dataset on a two-layer network architecture

VERSION	METRIC PARAMS	ACCURACY	AUC	DP
UNMOD.	0, 0	0.7818	0.6190	0.0478
PRE	0, 0	0.6937	0.5986	0.2773
POST	0, 0	0.7467	—	0.0365
FIRST	-100, 0	0.7769	0.6176	0.0440
	-1000, 0	0.7571	0.6190	0.0407
SECOND	-100, 1	0.7849	0.5870	0.0308
	-100, 10	0.7827	0.6165	0.0446
	-1000, 1	0.7076	0.6086	0.0343
	-1000, 10	0.7567	0.6070	0.0383

Table 5: Results for the Wine Quality dataset on a single-layer network architecture

VERSION	METRIC PARAMS	ACCURACY	AUC	DP
UNMOD.	0, 0	0.7100	0.7160	0.1266
PRE	0, 0	0.6793	0.6820	0.1040
POST	0, 0	0.6423	—	0.0592
FIRST	100, 0	0.7300	0.7322	0.1181
	1000, 0	0.7200	0.7239	0.1027
SECOND	100, 1	0.7162	0.7173	0.0891
	100, 10	0.7085	0.6605	0.0068
	1000, 1	0.7177	0.7212	0.1161
	1000, 10	0.7031	0.6549	0.0906

Table 6: Results for the Recidivism dataset on a single-layer network architecture

VERSION	METRIC PARAMS	ACCURACY	AUC	EO
UNMOD.	0, 0	0.6667	0.6664	0.2733
PRE	0, 0	0.5824	0.5916	0.0401
POST	0, 0	0.6149	—	0.1815
FIRST	100, 0	0.6733	0.6726	0.3206
	1000, 0	0.6591	0.6590	0.2823
SECOND	100, -1	0.6714	0.6713	0.2918
	100, -10	0.5928	0.5961	0.1084
	1000, -1	0.6676	0.6678	0.2690
	1000, -10	0.6013	0.6042	0.1503