GRAD: GRADIENT REVERSAL AGAINST DISCRIMINATION

//// // / / /// // //// / // //// // / A FAIR NEURAL NETWORK LEARNING APPROACH //// // / / /// // //// / // //// // /

Edward Raff & Jared Sylvester

Booz Allen Hamilton, Strategic Innovation Group

ICML Workshop on Fairness, Interpretability & Explainability, 13–15 July 2018, Stockholm

Abstract

No methods currently exist for inducing fairness in arbitrary neural network architectures. In this work we introduce GRAD, a new and simplified method for producing fair neural networks that can be used for auto-encoding fair representations or directly with predictive networks. It is easy to implement and add to existing architectures, has only one (insensitive) hyper-parameter, and provides improved individual and group fairness. We use the flexibility of GRAD to demonstrate multi-attribute protection.

Contributions

- *Goal*: build a network which is fair with respect to some protected attribute a_p.
  - i.e. output $\hat{y}$ given input x is invariant to a_p
- *Solution*: Gradient Reversal Against Discrimination
  - Simple to implement
  - Requires only one (insensitive) hyper-parameter
  - Applicable to any architecture, including:
    - Autoencoder architectures
    - Direct predictive architectures
  - Competitive with other approaches
  - The first neural network shown to protect multiple attributes concurrently

Method

- Neural network architecture with two sets of outputs
  - Inspired by domain adaptation (Ganin et al., '16)
- Several feature extraction layers (the "trunk"), followed by split into two "branches":
  - "*Target branch*" learns to predict target y
  - "*Attribute branch*" learns to predict protected attribute a_p
- Target branch can be either an autoencoder (GRAD–Auto) or a classifier/regressor (GRAD–Pred)
  - Completely architecture agnostic
  - We used 2 fully-connected layers in each branch
    - No architecture search performed
- Loss of network is sum of loss of each branch
  - Weighted by balancing parameter λ
    $$\ell(y, a_p) = \ell_t(y) - \lambda \cdot l_p(a_p)$$
- Losses for both $\ell_t(y)$ and $\ell_p(a_p)$ are calculated and gradients are used for weight updates as normal...
- *Except: once propagated down the attribute branch, gradients are reversed (i.e. multiplied by -1) before being applied to the trunk.*
- Effect:
  - Network can still accurately predict target
  - Network moves *away* from optima in predictions of a_p
    - Enforces ignorance of protected attribute

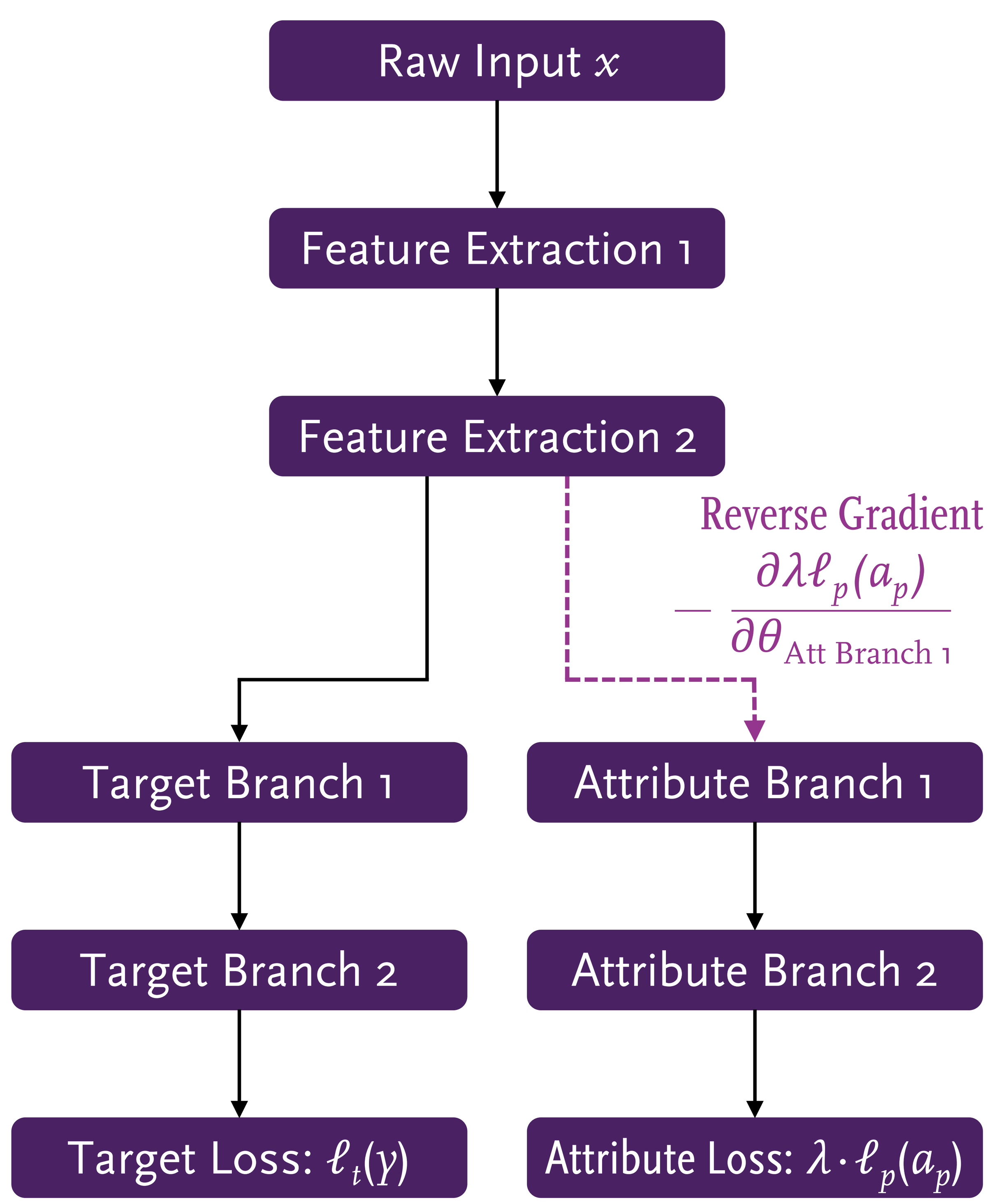


Diagram of the GRAD architecture. The dotted, purple connection indicates normal forward propagation but backpropagation with reversed signs. The value x is the input to the network, and the two terminal nodes are the losses that get backpropagated.

Evaluation

- Following Zemel et al., 2013
- Evaluated on German, Adult, Health & Diabetes
- *Discrimination* (a.k.a. "group fairness")
  - Difference between average predicted scores for each protected attribute value
- *Consistency* (a.k.a. "individual fairness")
  - Similar inputs should receive similar outputs
- *Accuracy*
- *Delta* = Accuracy – Discrimination

Results

- GRAD is typically best or 2nd best in each metric
- Competitive with prior methods in all metrics
- Both GRAD–Auto & GRAD–Pred reliably produce very high Consistency scores
  - One of the two is always the best in Consistency
- Capable of achieving Discrimination=0.00 & Consistency=1.00

Multiple Sensitive Attributes

- Sensitive attributes such as race & gender often co-occur in the same data set
  - No prior work has rigorously examined protecting multiple attributes concurrently
- GRAD can protect multiple attributes at once
- Protecting only a single attribute causes decreased fairness with respect to the other attribute

			Discrimination		
Algorithms	Acc	Delta	Race	Gender	Cons
NN-Auto	0.5735	0.5392	0.0412	0.0275	0.6411
GRAD-Auto	0.5765	0.5723	*0.0055*	**0.0030**	0.6288
NN-Pred	**0.6286**	*0.5848*	0.0418	0.0458	*0.6464*
GRAD-Pred	*0.5980*	**0.5949**	**0.0028**	*0.0034*	**0.7180**
GRAD-Auto-R	0.5851	0.5749	0.0003	0.0201	0.6404
GRAD-Auto-G	0.5640	0.5143	0.0981	0.0013	0.6093
GRAD-Pred-R	0.5844	0.5478	0.0020	0.0713	0.7538
GRAD-Pred-G	0.5941	0.5526	0.0785	0.0045	0.6849

Results on the Diabetes dataset. NN- models are standard neural networks. GRAD–Auto and GRAD–Pred protect both race & gender. *-R models protect only race and *-G models protect only gender.*

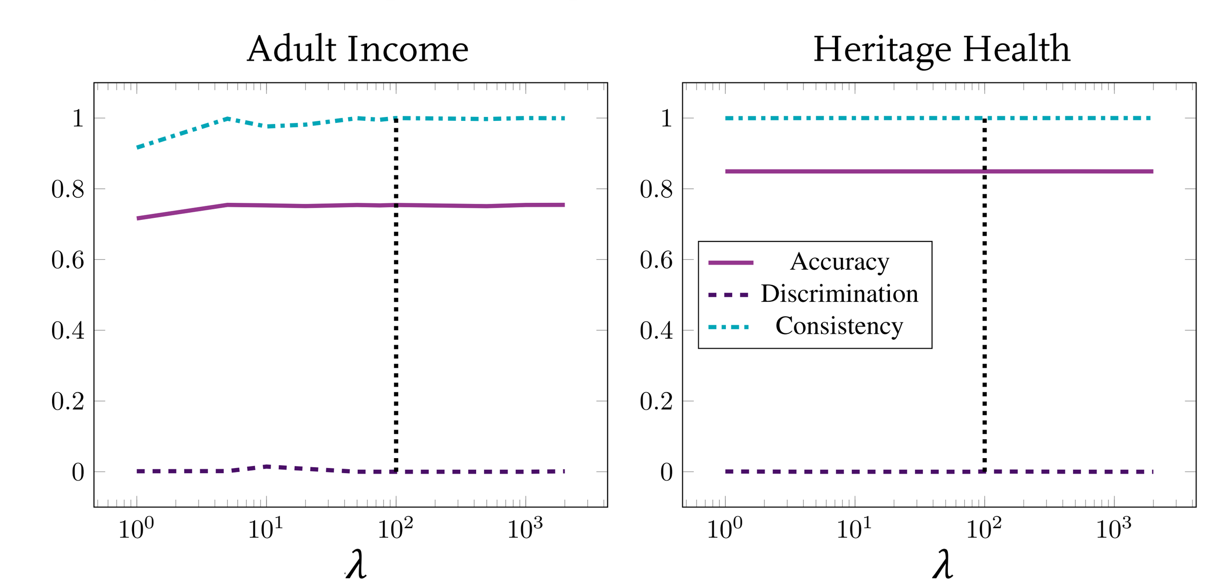
(Our techniques)

Algorithm	Acc	Delta	Discr	Cons
NN-Auto	**0.8506**	0.7939	0.0567	0.9730
GRAD-Auto	0.8491	**0.8491**	**0.0000**	**1.0000**
NN-Pred	0.8440	0.7511	0.0929	0.9453
GRAD-Pred	*0.8493*	0.8486	*0.0007*	*0.9999*
NBF	0.6878	0.5678	0.1200	0.5893
FF	0.8474	0.8474	**0.0000**	**1.0000**
LR	0.7547	0.6482	0.1064	0.7233
LRF	0.7212	0.7038	0.0174	0.6223
LFR	0.7365	0.7365	**0.0000**	**1.0000**
VFAE	0.8490	*0.8490*	**0.0000**	—

Results on the Heritage Health dataset. Best results in bold, second best in italics. NN = standard neural nets; NBF = fair Naïve Bayes; FF=Fair Forests; LR = Logistic Regression; LRF = fair Logistic Regression; LFR = Learned Fair Representations; VFAE = Variational Fair Autoencoders.

Robustness to λ

- λ controls trade-off between goals of predicting y and not predicting a_p
- Didn't do any hyper-parameter search for λ
  - Used λ=100 for all experiments
  - Keeps things simple for practitioners
- Any values in [20,1000] would have been acceptable



Performance of GRAD–Pred as a function of λ (on the x-axis; log scale). The y-axis shows Accuracy, Consistency (dotted; higher is better) & Discrimination (dotted; lower is better). The vertical, dotted black line shows the value λ = 100 used in all experiments.

Booz | Allen | Hamilton