FairPANs - bringing fairness to neural networks

AUTHORS

Hubert Ruczyński
Jakub Wiśniewski

ABSTRACT

The main topic of this study is the implementation and further research in the area of obtaining fair tabular data classifiers with the use of neural networks. To achieve such results, we modify the idea of GANs (Generative Adversarial Networks) by swapping the generator with the classifier and adapting the adversarial to recognize the label of a sensitive value. This way, PAN (Predictive Adversarial Network) should bring us much fairer predictions. As a result, we present a mitigation technique suitable for neural networks and explore this field even more.

INTRODUCTION TO FAIRNESS

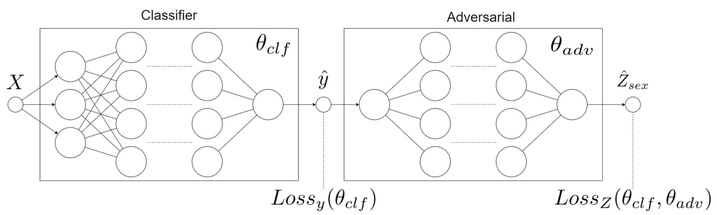
Consider the idea of the algorithm that has to predict whether giving credit to a person is risky or not. It is learning on real data of giving credits which were biased against females (historical fact). In that case, the model learns this bias, which is not only included in the simple sex variable but also is hidden inside other variables. Fairness enables us to detect such bias and handles a few methods to fight it. To learn more, I recommend the article 'Fairmodels: A Flexible Tool For Bias Detection, Visualization, And Mitigation' by Jakub Wisniewski and Przemysław Biecek.

INTRODUCTION TO GANS

Generative Adversarial Networks are two neural networks that learn together. The Generator has to generate new samples that are indistinguishable from original data and the adversarial has to distinguish if the observation is original or generated. The generator is punished whenever the adversarial makes the correct prediction. After such process generator eventually learns how to make indistinguishable predictions and adversaries' accuracy drops up to 50% when a model cannot distinguish the two classes.

MEET FAIRPANS

FairPANs are the solution to bring fairness into neural networks. We mimic the GANs by subsetting generator with classifier (predictor) and adversarial has to predict the sensitive value (such as sex, race, etc) from the output of the predictor. This process eventually leads the classifier to make predictions with indistinguishable sensitive values. The idea comes from blogs: Towards fairness in ML with adversarial networks (Stijn Tonk) and Fairness in Machine Learning with PyTorch (Henk Griffoen) however, our implementation in R offers slightly different solutions.



CUSTOM LOSS FUNCTION

The crucial part of this model is the metric we use to engage the two models into a zero-sum game. This is captured by the following objective function:

$$min_{\theta_{clf}}[Loss_y(\theta_{clf}) - \lambda Loss_Z(\theta_{clf}, \theta_{adv})]$$

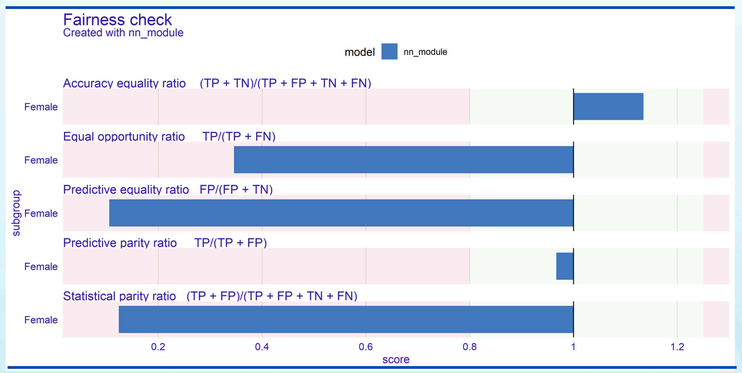
So, it learns to minimize its prediction losses while maximizing that of the adversarial (due to lambda being positive and minimizing a negated loss is the same as maximizing it). The objective during the game is simpler for the adversarial: predict sex based on the income level predictions of the classifier. This is captured in the following objective function:

$$min_{\theta_{clf}}[Loss_Z(\theta_{clf}, \theta_{adv})]$$

The adversarial does not care about the prediction accuracy of the classifier. It is only concerned with minimizing its prediction losses.
Firstly we pretrain classifier and adversarial. Later we begin the proper PAN training with both networks: we train the adversarial, provide its loss to the classifier, and after that, we train the classifier. This method shall lead us to fair predictions of the FairPAN model.

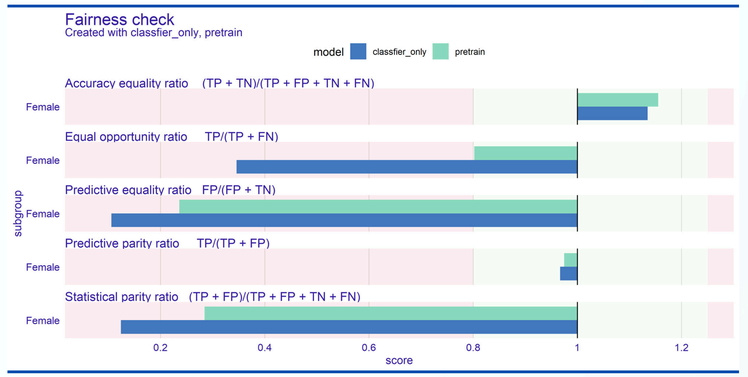
RESULTS

To show, that this method is reasonable and works properly, let us introduce you to a simple example of how the FairPAN works. The graph below represents the 5 most important fairness metrics for the adult data set, just after pretrain. For each of the metrics, which are explained in the plot, we calculate a ratio between the two classes. We assume that the model is fair according to the metric when it satisfies the 4/5 rule (the bar isn't on the red background). The 4/5 rule is satisfied when both labels' in a sensitive class get similar results (ratio between 4/5 and 5/4). Let's note that the most important for us to minimize it is the STP ratio (Statistical Parity Ratio).

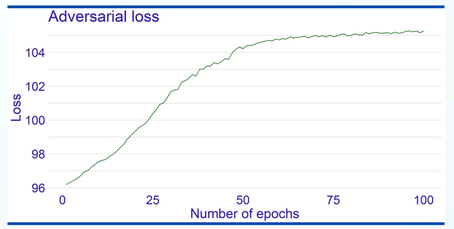


As we can see, the STP ratio after the pretrain is really bad, which means that there is a lot to improve in this case.

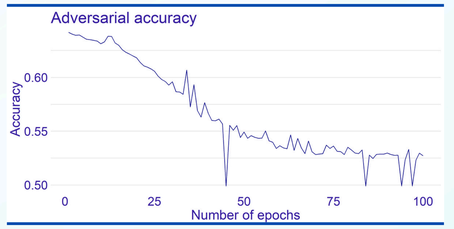
To accurately show the progress made by FairPAN, the next plot represents the metrics for the classifier model after 50 epochs of training. As we can see, during the training it worsens its fairnessmetrics in order to optimize its acc



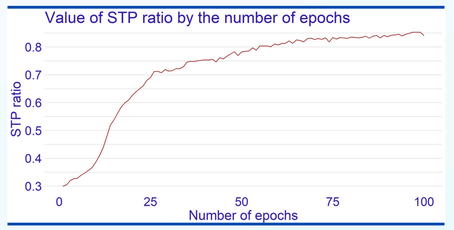
To check if our model works properly, we've decided to track the three most important metrics which are: loss and accuracy of the adversarial model and of course STPR itself. These statistics indicate if the model is learning well and makes desirable progress. [1][2] Loss function should grow and accuracy should diminish because it means that adversarial is worse with every iteration. It means that the classifier makes more fair predictions (the adversary can't say which label is which). [3] STPR in this case should grow towards the 0.8-1.25 range which means that predictions are fairer. Moreover, we decided to monitor the accuracy of the classifier [4].



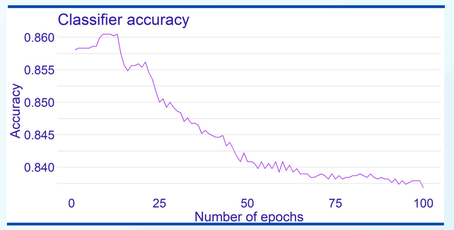
[1] Loss of the adversarial model



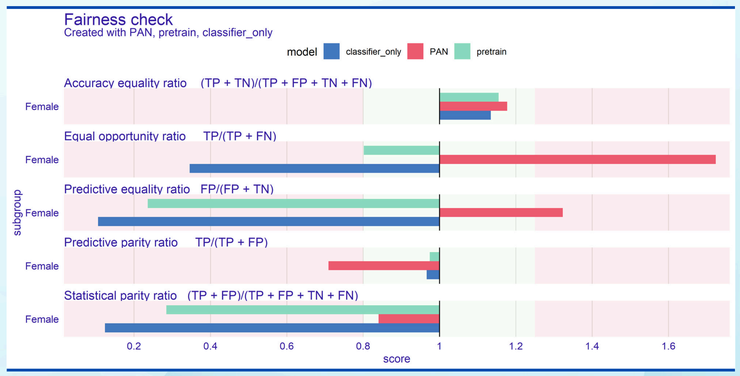
[2] Accuracy of the adversarial model



[3] STP ratio for PAN model



[4] Accuracy of PAN model



As we can see from the last plot, FairPAN training led to significant STP ratio improvement, so that the model became fair according to this metric. It isn't absolutely costless, because other fairness metrics worsened however, it is impossible to improve one metric without worsening the others. As you can see from the table below, performance metrics also diminished, but the loss is pretty low in comparison to ordinary mitigation techniques and we lose only 1.3% of the accuracy to make our model fair.

	Accuracy	AUC	F1	Precision	Recall
Classifier	0.850	0.871	0.541	0.773	0.417
PAN	0.837	0.857	0.536	0.680	0.443