Fairness: Mitigating bias

Once a source of bias has been identified in the training data, we can take proactive steps to mitigate its effects. There are two main strategies that machine learning (ML) engineers typically employ to remediate bias:

- Augmenting the training data.
- Adjusting the model's loss function.

Augmenting the training data

If an audit of the training data has uncovered issues with missing, incorrect, or skewed data, the most straightforward way to address the problem is often to collect additional data.

However, while augmenting the training data can be ideal, the downside of this approach is that it can also be infeasible, either due to a lack of available data or resource constraints that impede data collection. For example, gathering more data might be too costly or time-consuming, or not viable due to legal/privacy restrictions.

Adjusting the model's optimization function

In cases where collecting additional training data is not viable, another approach for mitigating bias is to adjust how loss is calculated during model training. We typically use an optimization function like log loss to penalize incorrect model predictions. However, log loss does not take subgroup membership into consideration. So instead of using log loss, we can choose an optimization function designed to penalize errors in a fairness-aware fashion that counteracts the imbalances we've identified in our training data.

The TensorFlow Model Remediation Library provides utilities for applying two different bias-mitigation techniques during model training:

- **MinDiff**: MinDiff aims to balance the errors for two different slices of data (male/female students versus nonbinary students) by adding a penalty for differences in the prediction distributions for the two groups.
- **Counterfactual Logit Pairing**: Counterfactual Logit Pairing (CLP) aims to ensure that changing a sensitive attribute of a given example doesn't alter the model's prediction for that example. For example, if a training dataset contains two examples whose feature values are identical, except one has a `gender` value of `male` and the other has a `gender` value of `nonbinary`, CLP will add a penalty if the predictions for these two examples are different.

The techniques you choose for adjusting the optimization function are dependent on the use cases for the model. In the next section, we'll take a closer look at how to approach the task of evaluating a model for fairness by considering these use cases.

Exercise: Check your understanding

Which of the following statements regarding bias-mitigation techniques are true?

- [ ] Adding more examples to the training dataset will always help counteract bias in a model's predictions.
- [x] MinDiff penalizes differences in the overall distribution of predictions for different slices of data, whereas CLP penalizes discrepancies in predictions for individual pairs of examples.

  MinDiff addresses bias by aligning score distributions for two subgroups. CLP tackles bias by ensuring that individual examples are not treated differently solely because of their subgroup membership.

  1 of 2 correct answers.
- [x] Both MinDiff and CLP penalize discrepancies in model performance tied to sensitive attributes

  Both techniques aim to mitigate bias by penalizing prediction errors resulting from imbalances in how sensitive attributes are represented in training data.

  2 of 2 correct answers.
- [ ] If you are mitigating bias by adding more training data, you shouldn't also apply MinDiff or CLP during training.

Key terms:

- Bias (ethics/fairness)
- Log Loss

Help Center

Previous
Identifying bias (10 min)

Next
Evaluating for bias (5 min)

Was this helpful?

Send feedback

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License, and code samples are licensed under the Apache 2.0 License. For details, see the Google Developers Site Policies. Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2024-10-09 UTC.