Adversarial De-Biasing

This approach requires a fundamental change to the structure of the model, and therefore is **not** necessarily the best approach. However, it is a valid one that I believe will likely have some positive impact. Think of this model as a neural network with one head towards predicting y, some significant aspect of the data, and another towards predicting z, some biased variable (race or age, for example). Consider $G(x)$ to be the shared embedding for the layer. Let $f(x)$ be some prediction function executed by the model, such that $y = f(g(x))$. Additionally, set a to be an adversarial layer, such that $z = a(g(x))$. The unification of these functions is done through some negative gradient defined N_λ, where λ is a tunable hyperparameter. The overall optimization, therefore, becomes:

$$min \left[\sum_{(x,y)\in X} L_y(f(g(x)) + \sum_{(x,z)\in S} L_z(a(N_\lambda(g(x)), z) \right]$$

where L_y and L_z are the respective loss functions for y and z. Additionally, the secondary approach could be considering the adversarial model to be wholly separate, which may reduce complexities in gradient updates. The adversarial model is solely focused on minimizing its own loss, ergo, its gradient weights U are updated by $\Delta_u L_a$. The predictor's weights, however, are calculated by

$$\Delta_w L_p - proj_{\Delta_w L_a} \Delta_w L_p - \alpha \Delta_w L_a$$

where α is a hyperparameter controlling accuracy/debiasing tradeoff. We could also seek to maximize the entropy of the predictive model, effectively playing a fully zero-sum game of predictive accuracy vs. bias removal.

Robust Optimization

Consider a population with some K groups, where each K composes some proportion α_K of the overall population with the probability distribution P_K. Assume neither the proportion nor underlying distribution are known. Some inputs (or values) associated with $z \sim P_K$ have some known risk (or loss) of $R_K(\theta)$, which composes the expected loss over those inputs. Ergo, this is the following:

$$\mathbb{R}_{max}(\theta) = max_{k\in K}\mathbb{R}_k(\theta),\ \mathbb{R}_k(\theta) := \mathbb{E}_{P_k}[l(\theta, z)]$$

where we control the worst-case risk over all K groups. This is not directly usable, because we do not know the identity (or the underlying P_k). Ergo, we switch gears a little bit, and consider all distributions which are reasonably close to our observed, overall population distribution. Represent this as some Q within chi-squared radius r, where:

$$\mathfrak{R}_{ro} := sup_{Q\in\mathfrak{B}}\ \mathbb{E}_Q[l(\theta; Z)]$$

and r is chosen from $r_{max} = (1/\alpha_{min} - 1)^2$. Intuitively, or at least I hope so, we are now minimizing for the worst case distribution that make up at least α_{min} percent of the overall population. Since this is a hyperparameter, we should be able to set some value for α_{min} and train a model (though I don't think it would necessarily look like this in its final form).