Transfer Learning to Identify Fair, Causal, or General Representations in Deep Neural Networks

Jacy Reese Anthis [1] [2]

[1]University of Chicago [2]Sentience Institute

Motivation

The inner parametric structure of deep neural networks remains largely a black box.

Figure 1. Viral tweets led the New York government to investigate Goldmach Sachs for algorithmic bias.

Goldman Sachs, the firm behind the Apple Card, defended itself by claiming its algorithm did not take protected classes such as gender as input. However, being **"gender-blind"** is not necessarily the right way to handle risks of discrimination. One reason for this is that other social variables (e.g., occupation, income, spending habits) can be used to predict protected classes.

In other areas such as causation and generalization, this interpretability is also valuable. More efficient, accurate, and sophisticated AI systems may require models of causal mechanisms and generalizations beyond a relatively narrow domain. **Can we provide an informative measure of whether a model has an internal representation of a certain variable?**

Previous work

- Zemel et al. [2013] developed the notion of a **fair representation** in machine learning, a dual optimization of data representation that maximizes information retained from the data while retaining no information about membership in the protected class.
- Chouldechova [2017], Corbett-Davies et al. [2017], and Kleinberg et al. [2016] show the **trade-offs** between different definitions of algorithmic fairness, such as demographic parity and equalized odds.
- There are many papers addressing fairness as an instance of the general goal of **invariant representation** [Edwards and Storkey, 2016, Zhang et al., 2018, Zhao et al., 2020a,b]. Invariant representations are also a way to operationalize the **generalization** of a model across domains [Anselmi et al., 2014, Ben-David et al., 2007, 2010, Ganin et al., 2016, Zhao et al., 2018, 2020c], differential privacy [Coavoux et al., 2018, Hamm, 2015, 2017], and even **causation** in which the model should be invariant to the counterfactual pathway that led to the data we possess [Johansson et al., 2018, 2021, Shalit et al., 2017].
- There are numerous tools for the interpretability of **image processing** neural networks, such as saliency maps and localization techniques [Baehrens et al., 2009, Simonyan et al., 2014, Selvaraju et al., 2017]. These mappings can be constrained such that the changed image stays within a manifold determined by the original dataset [Miller et al., 2019, Nguyen et al., 2016], and a common example of such a manifold is a **Generative Adversarial Network (GAN)**, which produces output that can help interpret the generative network [Chen et al., 2016, Kobayashi et al., 2017].

Problem Setup

We take the typical machine learning problem with an input dataset X and output variable Y. The goal is to estimate a prediction function $f : \mathcal{X} \mapsto \mathcal{Y}$ that minimizes $\mathbb{E}[\ell(f(X), Y)]$ for some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$. There is a third random variable A (e.g., race and gender in fairness), and we aim to provide a metric for the extent to which information about A is retained in a model's representation $Z = g(X)$.

Transfer Learning Measures of Inner Representation

We propose to measure inner representation by (**a**) the performance of a retrained neural network (frozen except the final layer) on the prediction of the sensitive characteristic:

$$a = \ell(A, \hat{A})$$

to (**b**) transfer learning performance on prediction of the sensitive characteristic relative to the prediction of m other variables $B_1, ..., B_m$ masked from trainining in the source domain $\mathcal{D}$,:

$$b = \frac{\ell(A, \hat{A})}{\frac{1}{m}\sum_1^m \ell(B, \hat{B})}$$

We can measure (**c**) performance relative to the performance of a uninomial or multinomial logistic regression trained on the input data, with loss denoted $\hat{A}_{LM}$:

$$c = \frac{\ell(A, \hat{A})}{\ell(A, \hat{A}_{LM})}$$

or (**d**) condense the model's representation of sensitive characteristics by freezing at a low-dimensional bottleneck layer, with bottleneck loss denoted $\hat{A}_{BN}$:

$$d = \ell(A, \hat{A}_{BN})$$

We can also form 4 additional metrics based on permutations: (**bc**) performance relative to other variables and logistic regression, where weights $\alpha + \beta = 1$ are the relative importance of the two terms in the denominator:

$$bc = \frac{\ell(A, \hat{A})}{\alpha\ell(A, \hat{A}_{LM}) + \frac{\beta}{m}\sum_1^m \ell(B, \hat{B})}$$

(**bd**) performance with a bottleneck layer relative to other variables:

$$bd = \frac{\ell(A, \hat{A}_{BN})}{\frac{1}{m}\sum_1^m \ell(B, \hat{B})}$$

(**cd**) performance with a bottleneck layer relative to logistic regression.

$$cd = \frac{\ell(A, \hat{A})}{\ell(A, \hat{A}_{LR})}$$

and (**bdc**) performance with a bottleneck layer relative to other variables and logistic regression:

$$bcd = \frac{\ell(A, \hat{A}_{BN})}{\alpha\ell(A, \hat{A}_{LR}) + \frac{\beta}{m}\sum_1^m \ell(B, \hat{B}_{BN})}$$

Each of these metrics can also be normalized for comparison.

Simulation Results

We test these measures in the simplest neural network capable of forming such representations, one inner layer with two inner nodes and an output layer with one node. This allows the model to learn a data-generating process where A is based on an XOR relationship between x_1, x_2, B is based on an XOR relationship between x_3, x_4, and Y is linearly based on A, each with random noise.

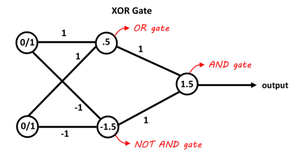

Figure 2. Example of XOR implemented in a neural network (source).

We find measures of (a) 0.36, (b) 0.72, (c) 0.72, (bc) 0.72, in which a higher measure corresponds to less representation of A within the network. We do not calculate the (d) measures for this because we have no bottleneck layer. We try a range of different simulation hyperparameters. While these largely show expected results, we do see in some cases that even this highly contrained network structure can still be retrained onto B with better accuracy than chance:

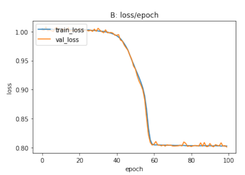

Figure 3. Sometime the model is able to be retrained to predict B, but only after many epochs.

Why is this? If the 2 inner nodes happen to have appropriately signed weights for producing the $XORx_3, x_4$ function that produces B, then it is possible for the nodes to be repurposed by the output layer for estimation of B. This indicates an important challenge in larger networks, given the fact that nodes are usually not reduced to 0 (as in LASSO regression), so a sufficiently large network technically has some representation of a vast range of nonlinear combinations of its input data. Thus it may be preferable for the inner representation measure to account for the ease of transfer learning onto A and B.

Future Plans

I plan to develop these measures in more depth and run tests on real-world data. I may also switch to a machine vision context, given the usefulness of GANs in manifesting the representation space. I am also interested in training language models to produce "thought" text, where they are trained on human-provided explanations of text prediction. In either case, I hope to make this my third-year research paper required by the Econometrics & Statistics department.