Capstone: Enhancing Performance, Maintaining Interpretability in the Social Sciences with Deep Learning

Hong Zheng

May 2024

Abstract

In the realm of social sciences, the quest for robust analytical tools that balance predictive accuracy with clarity and understandability has led to the exploration of innovative methodologies and techniques. Among the approaches is the Double Machine Learning (DML) method, which is used to estimate the causal effect of a treatment variable D on an outcome variable Y. It works by first removing the confounding effects of covariates, X, from both the treatment, D, and the outcome variables Y through machine learning models, allowing for a more accurate estimation of the treatment effect by focusing on the residual variations that are not explained by the covariates. We aim to integrate deep learning within the DML framework to harness its computational power to manage the nonlinear relationships often present in social science data. We employ this technique on a regression task on the `Communities and Crime` dataset and on a classification task on the `Adult` dataset. The performances of this approach on both tasks do not exceed that of benchmark DML architectures using non-deep-learning models.

Algorithm 1 Estimation of Treatment Effect using Double Machine Learning

1: **Input:** Data $\{(X_i, Y_i, D_i)\}_{i=1}^n$, where X_i are covariates, Y_i is the outcome, and D_i is the treatment indicator.
2: **Output:** Estimated treatment effect $\hat{\tau}$.

3: Train a model to predict the treatment D from covariates X. Let $f_D(X)$ be the estimated treatment propensity score.
4: Train a model to predict the outcome Y from covariates X and the treatment D. Let $f_Y(X, D)$ be the estimated outcome model.

5: Calculate the residuals: $\hat{U}_i = Y_i - f_Y(X_i, D_i)$, where $D_i = \text{argmax}_D f_D(X_i)$
6: Use the residuals $\hat{U}_i$ as the outcome in a regression model to estimate the treatment effect: $\hat{\tau} = \text{argmin}_\tau \sum_{i=1}^n \left(\hat{U}_i - \tau \cdot D_i\right)^2$

		First Model	
		Non-DL	**DL**
Second Model	**Non-DL**	Non-DL & Non-DL	DL & Non-DL
	DL	Non-DL & DL	DL & DL

Table 1: Model combinations

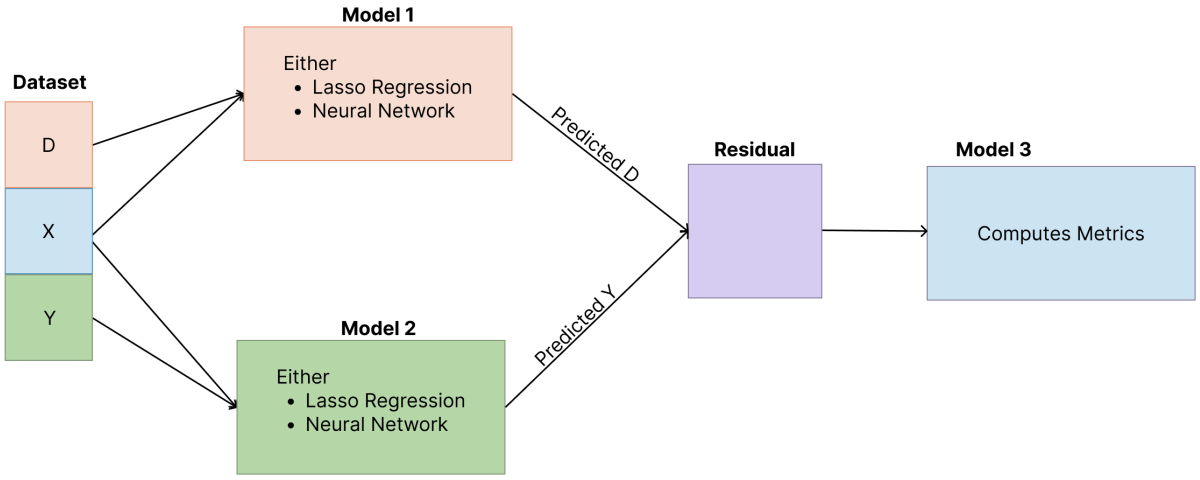

Figure 1: Double machine learning framework

Treatment Model	**Outcome Model**	**Effect Model**	**MSE Loss**
non-DL (Linear Regression)	non-DL (Linear Regression)	non-DL (Linear Regression)	0.025466
non-DL (Lasso)	non-DL (Lasso)	non-DL (Linear Regression)	0.051310
non-DL (Lasso)	DL (Neural Network)	non-DL (Linear Regression)	0.026257
DL (Neural Network)	non-DL (Lasso)	non-DL (Linear Regression)	0.048934
DL (Neural Network)	DL (Neural Network)	non-DL (Linear Regression)	0.027830
DL (Neural Network)	DL (Neural Network)	DL (Neural Network)	0.027829

Table 2: Regression task results

Treatment Model	**Outcome Model**	**Effect Model**	**Accuracy**	**F1 Score**	**Log Loss**
non-DL (Logistic)	non-DL (Logistic)	non-DL (Logistic)	0.844887	0.684506	0.324931
non-DL (Logistic)	DL (Neural Network)	non-DL (Logistic)	0.833720	0.650395	0.971046
DL (Neural Network)	non-DL (Logistic)	non-DL (Logistic)	0.844887	0.684506	0.450457
DL (Neural Network)	DL (Neural Network)	non-DL (Logistic)	0.823991	0.642568	1.045052
DL (Neural Network)	DL (Neural Network)	DL (Neural Network)	0.824323	0.642520	0.659059
non-DL (Logistic)	non-DL (Logistic)	DL (Neural Network)	0.850525	0.660131	5.185066

Table 3: Classification task results