Regression Adjustment with Artificial Neural Networks

Vinci Chow
The Chinese University of Hong Kong

November 1st 2016

Motivation

- Age of Big Data: data comes in a rate and in a variety of types that exceed our ability to analyse it
  - Texts, image, speech, video…
- Real motivation: a project Travis and I are working on studying whether an “afternoon effect” exists in local license plate auctions
  - Value of license plates mostly depend on aesthetic and superstitious factors
  - E.g. **168** would be more valuable than **861**
  - Hard to study. Previous studies resort to a large number of dummies

Regression Adjustment

- When treatment assignment correlates with untreated outcome, simple difference between treatment and control outcomes cannot estimate the true effect
  - In the aforementioned case, we cannot be sure that valuable plates were evenly divided between morning and afternoon auction sessions

- **Regression adjustment** is one of the standard remedy
  - Fit two models, one for treatment and one for control
  - The models are used to generate counterfactual outcomes
  - The estimated treatment effect is the average difference between observations' real and counterfactual outcomes

Main Idea

- Estimating the value of a plate is essentially a *translation* task
  - Translate a bunch of letters and numbers to a value
- Use **machine learning** models that are known to work well in automated translation
  - Think Siri and Google Translate
- Issue: Statistical properties of such models are not necessarily suitable for hypothesis testing. Examples:
  - Ridge regression and lasso are biased
  - Bayes rule under the assumption of uncorrelated covariates (*Naïve Bayes*) is a decent discrete choice model but a poor estimator
- Use simulations to study the properties of using such models

Machine Learning

- Computer scientists have studied these data types under the umbrella of machine learning
  - This includes familiar statistically techniques such as regression and maximum likelihood
  - Less common (to econometricians) such as k-neighbors, regression trees and artificial neural networks
  - To statisticians, machine learning is kind of like econometrics—new names, not necessarily new stuff
  - Objective is usually accurate prediction, hypothesis testing is rare

- Artificial neural networks have been shown to work well with complex data

Artificial Neural Network

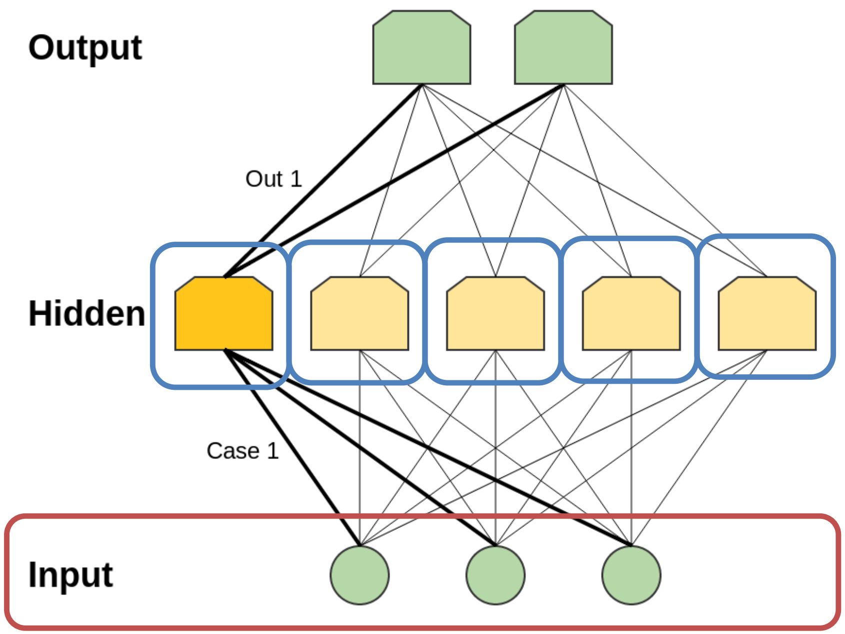
- Artificial Neural Networks are *biologically-inspired* models, consisting of interconnected neurons
- As a simple example, suppose each observation has three independent variables x_i
- The values of these three variables are fed to a number of **hidden neurons**, which combine them linearly and transform them with an **activation function** $F(\cdot)$

$$F\left(b_j + \sum w_{ji} x_i\right)$$

- The activation function is either logistic, tanh or most recently, *rectified linear unit*:

$$F(z) = \max(0, z)$$

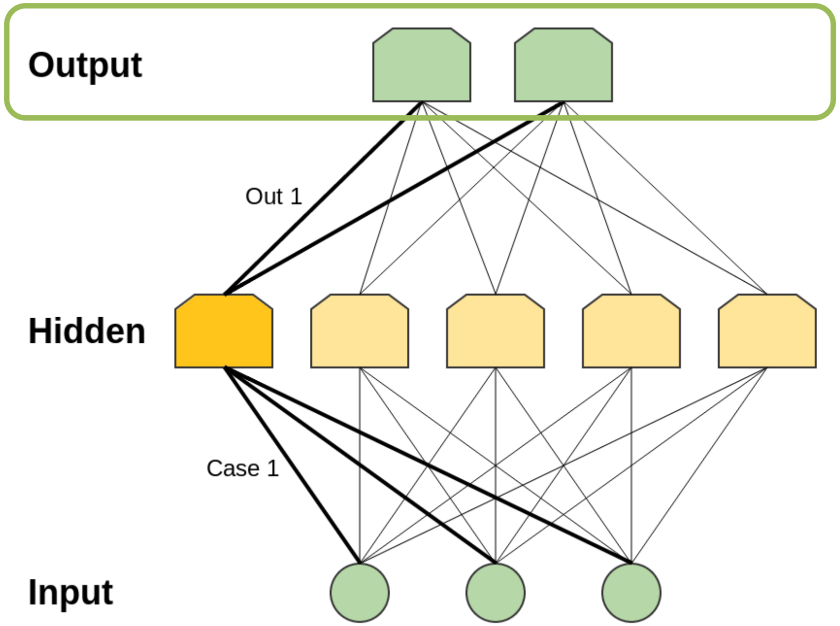
- b_j and w_{ji} need to be fitted



Source: colah's blog

Artificial Neural Network

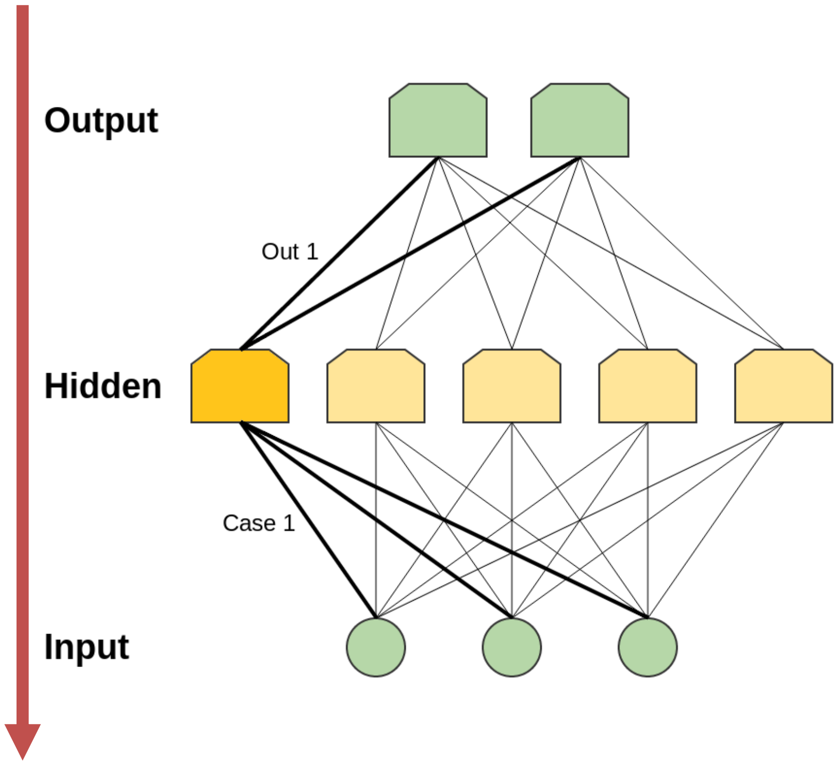
- The outputs from the hidden neurons are fed into the output neurons, which combine them linearly and transform them again
- The number of output neurons depends on the nature of the dependent variable
  - Single output neuron for linear or binary dependent variable
  - Multiple output neurons for categorical variable, each representing a score for a category. The outputs of all output neurons would be combined through a softmax function— i.e. multinomial logit



Source: colah's blog

Artificial Neural Network

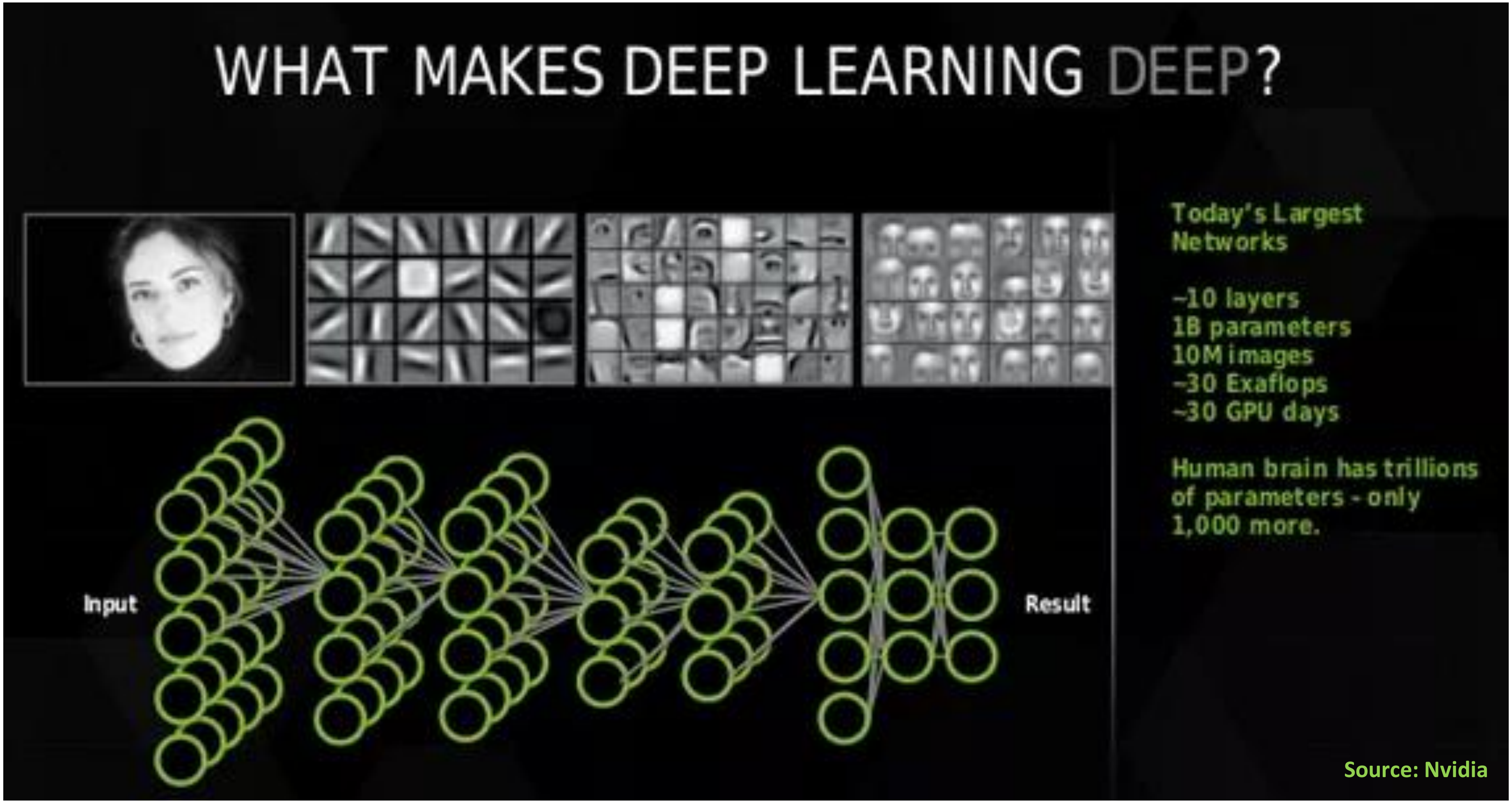
- Parameter estimate is conducted through **back propagation**
  - The residual $(\hat{y} - y)$ is used to correct the parameters in each layer through repeated use of chain rule
  - This process could become unstable as the number of layers increase
  - Techniques developed to overcome this problem: carefully chosen initial values, variable learning rates and normalize output values after every layer



Source: colah's blog

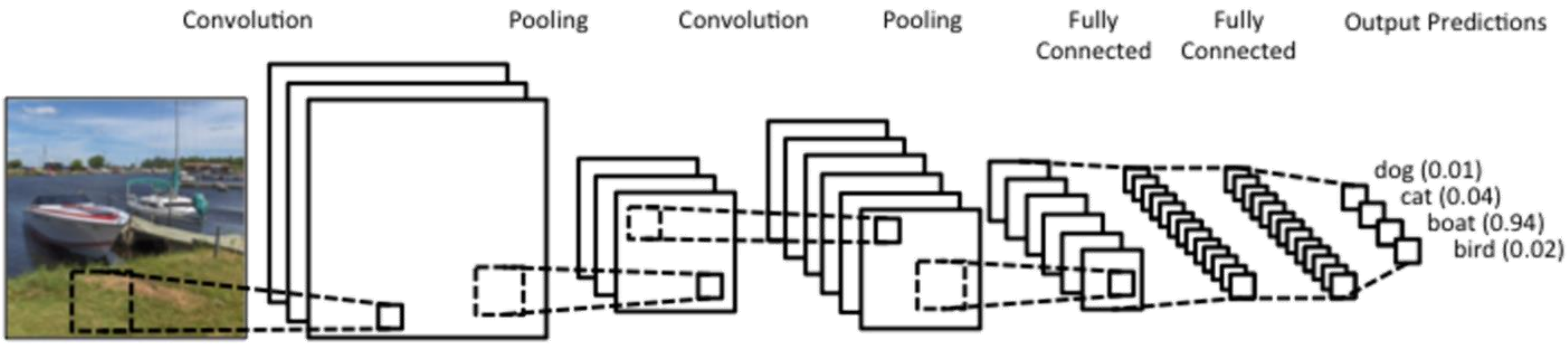
Deep Learning

- **Deep Learning** refers to the stacking of multiple hidden layers
  - Typically in the single digit, but can go as high as a hundred layers



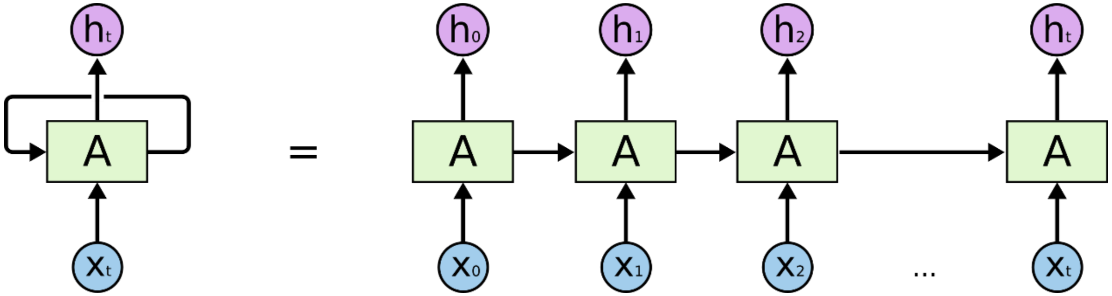
Different Types of ANN

- Convolutional Neural Networks
  - Each neuron is only connected to neighboring neurons



Source: WILDML

- Recurrent Neural Networks
  - Auto-regressive neurons with the ability to forget



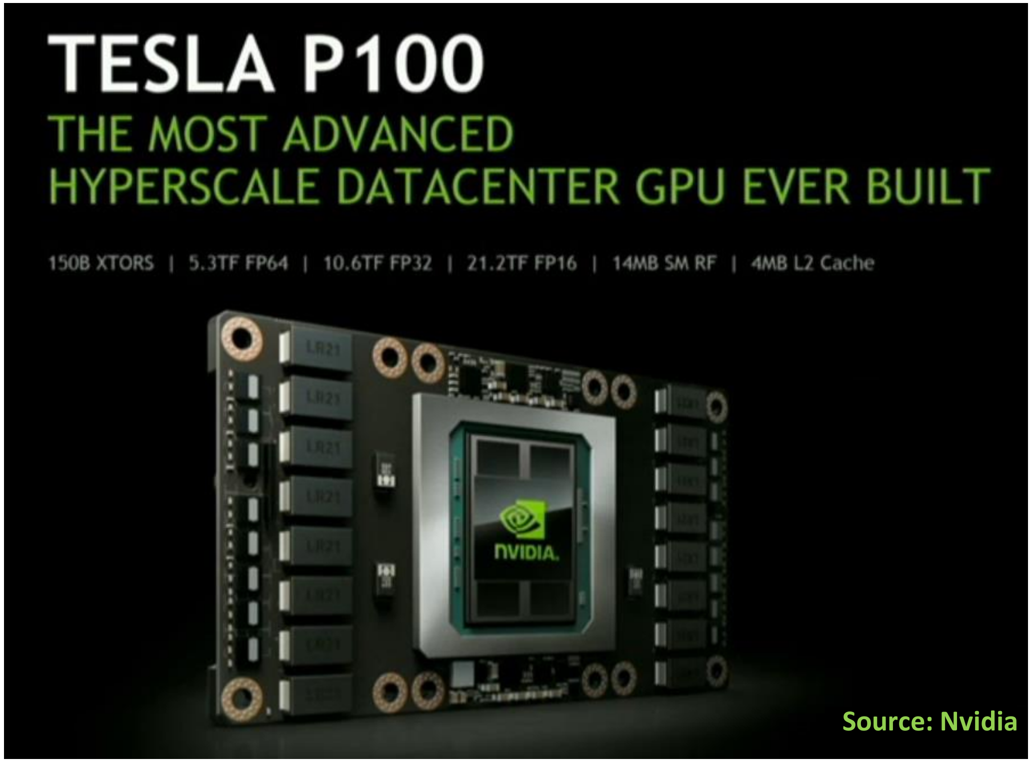
Source: colah's blog

Computation

- The idea of artificial neural network can be traced back to the 1940s
- Due to the large number of parameters and large data size involved, effective use of ANN is prohibitive until recently
- ANN took off in recent years due to massive increase in computational capabilities, particularly in the use of graphic processing unit (GPU) for computation

(3 variables + intercept) × 5 hidden neurons
= 20 parameters to fit

(30 variables + intercept) × 1000 hidden neurons
× 5 layers
= 155,000 parameters to fit



Hyperparameters

- The number of neurons per layer, the number and types of layers to use as well as the rate of learning has to be hand picked. These are called **hyperparameters**

- Hyperparameters are chosen through **cross validation**
  1. Separate data into 3 sets: train, validation and test
  2. The train set is used to train the model. This is repeated for every combination of hyperparameters
  3. The combination of hyperparameters that best predicts the validation set is chosen
  4. The test set is only used for reporting the goodness-of-fit of the chosen hyperparameters

Simulation

- Data-generating process is linear

$$y = \sum \beta_i x_i + \sum \delta_{jk} x_j x_k + \sum \gamma_{lmn} x_l x_m x_n + \cdots$$

  - 3000 samples (larger samples in progress)
  - No. of x: 10, 50, 100

    Multinomial distribution with uniformly distributed correlation
  - No. of high-order correlation: 0%, 50%, 100%, 200% of no. of x, ranging from 2^{nd} order to 4^{th} order
  - β, δ, γ: uniformly distributed
  - Mean 0 and SD 10

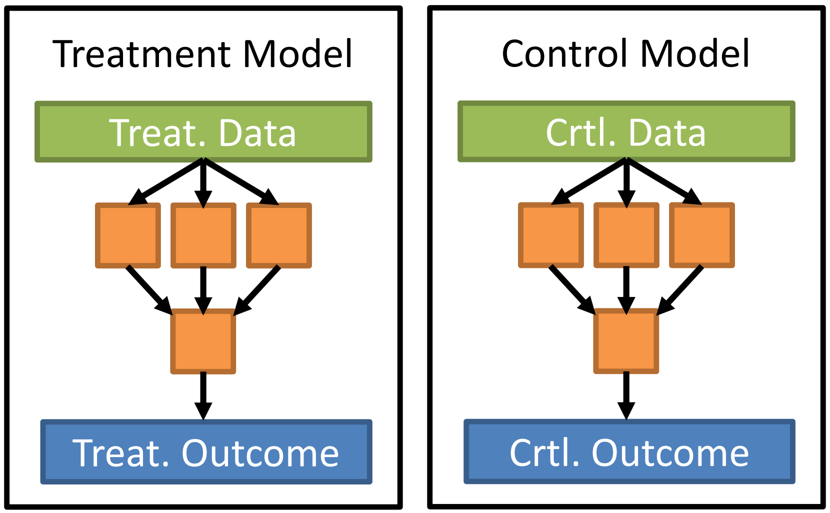
Simulation

- Treatment assignment
  - Baseline probability: 50%
  - Bonus for observations with y > 0 ranging from -20% to +20%
- Treatment effect
  - Ranges from -2 to +10
- Model
  - Layers: 1-4 layers
  - Number of neurons: 10 to 500
- 30 runs for every set of parameters. Report median estimates

Regression Adjustment + Neural Network

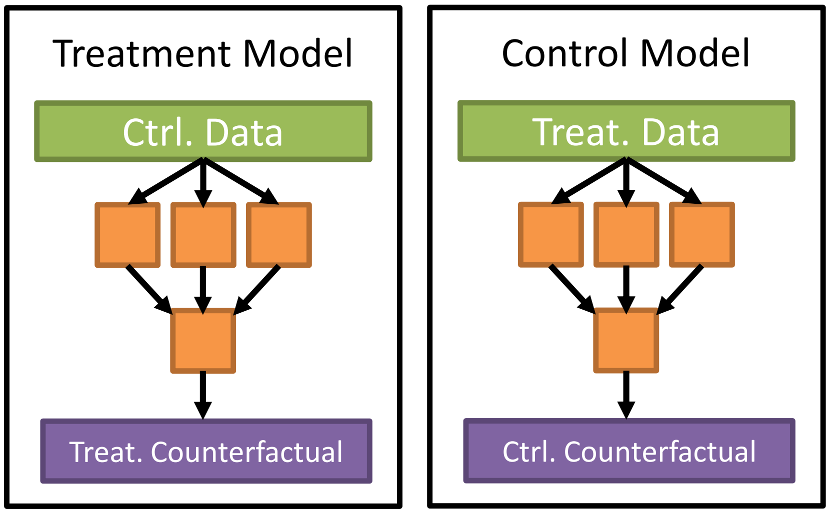
1. Training

Fit two models



2. Prediction

Generate Counterfactuals



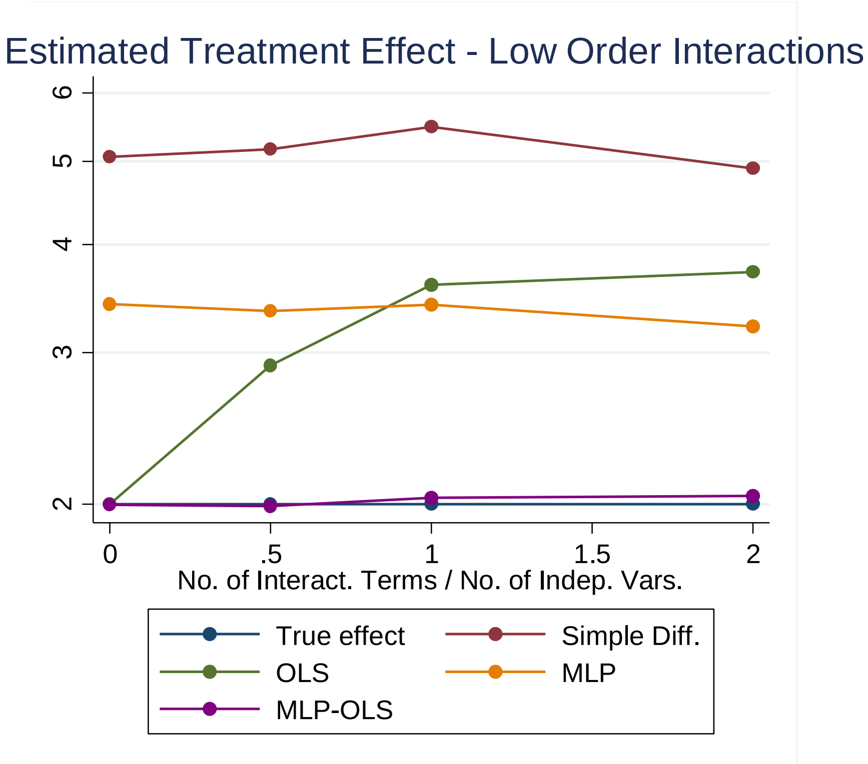
3. Regression Adjustment

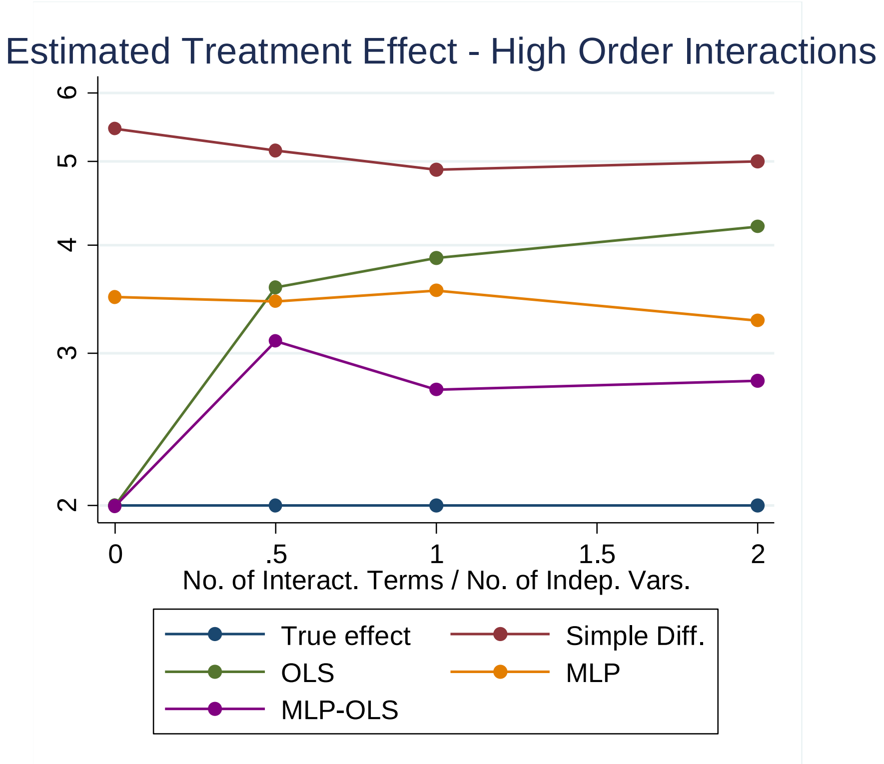
Estimate treatment effect

Treat. Outcome ∪ Treat. Counterfactual − Ctrl. Counterfactual ∪ Crtl. Outcome

How Well Does Neural Network Perform?

- Pure neural network (MLP) works better than OLS when the number of interaction terms is high





Settings: treatment effect=2, treatment chance bonus=0.2, single layer of 100 neurons

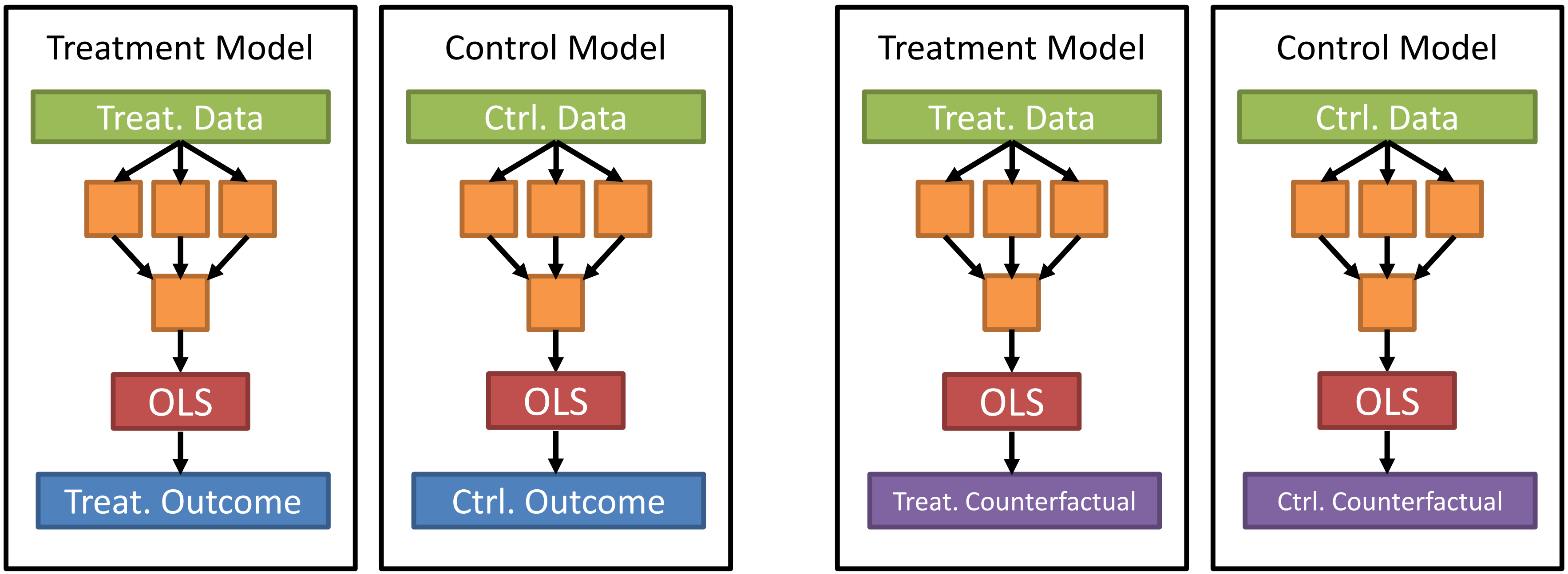
Improving Neural Network's Performance

1. Training

Fit two models

2. Prediction

Generate Counterfactuals

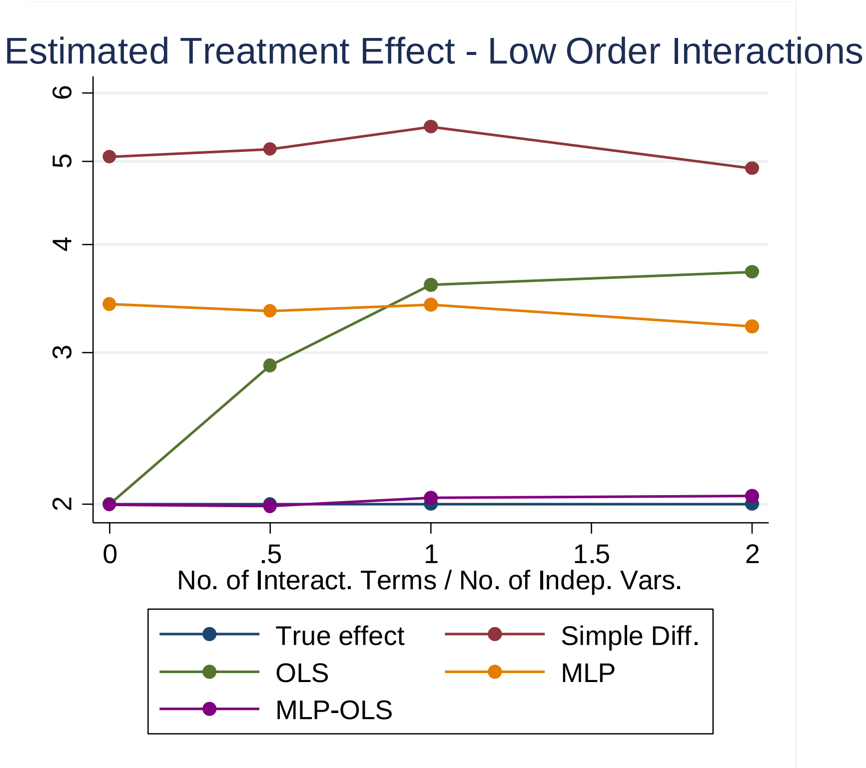


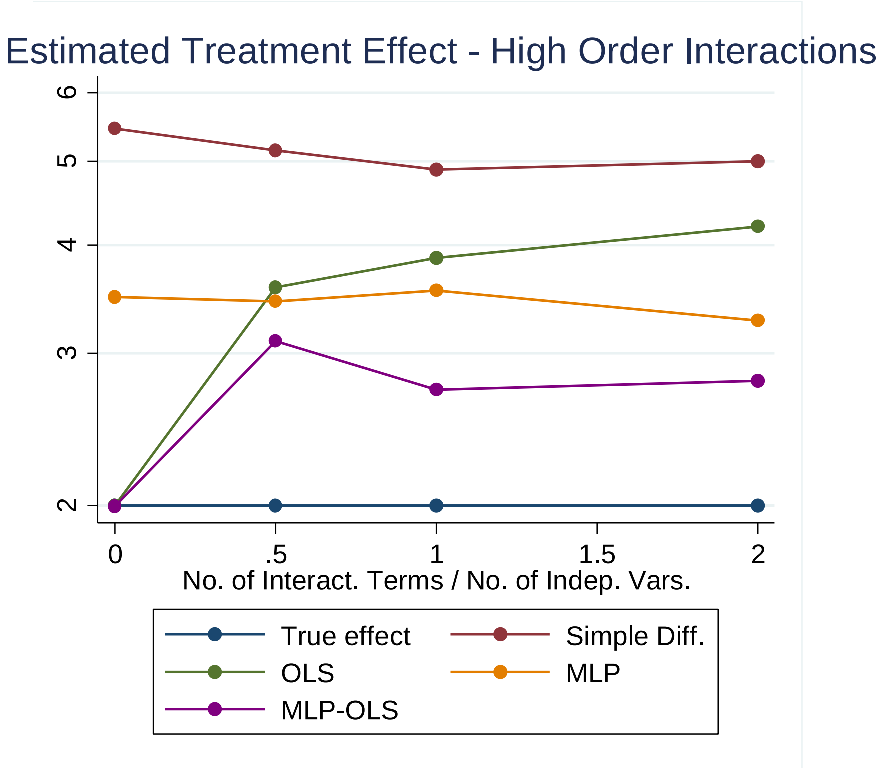
3. Regression Adjustment

Estimate treatment effect

How Well Does Neural Network Perform?

- Pure neural network (MLP) works better than OLS when the number of interaction terms is high
- Feeding the predicted values from a neural network into OLS works best (MLP-OLS)

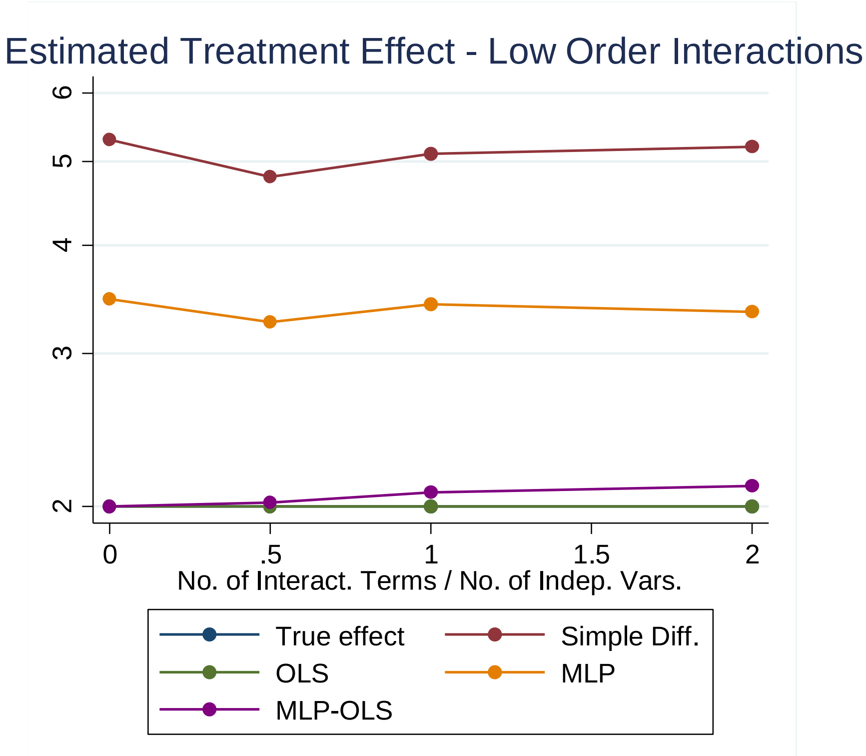


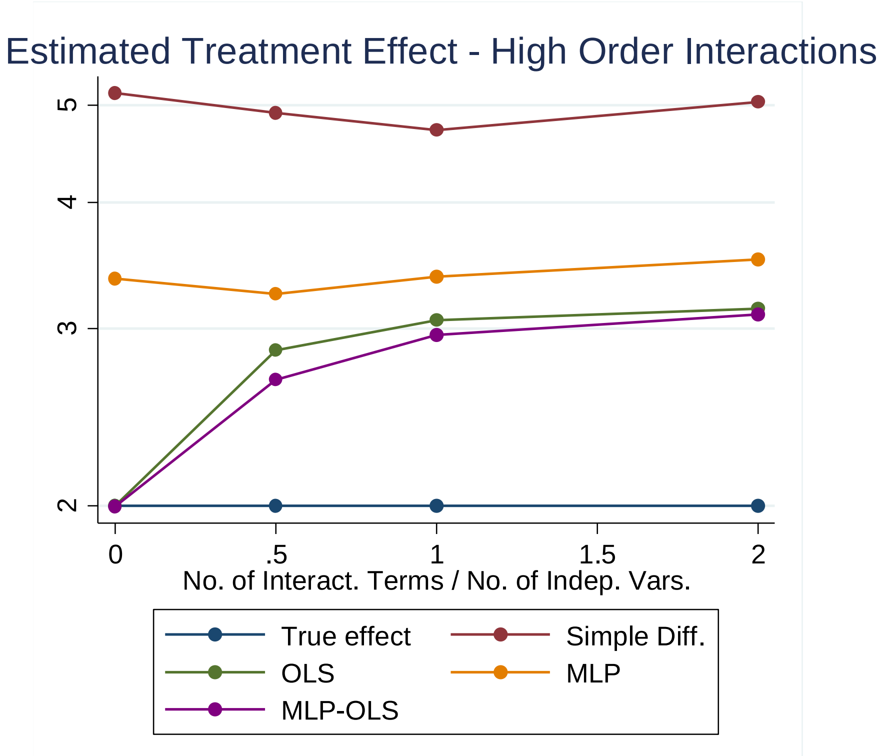


Settings: treatment effect=2, treatment chance bonus=0.2, single layer of 100 neurons

How Well Does Neural Network Perform?

- Adding 1st order interaction terms to OLS greatly improves its performance, but MLP-OLS is still better when high order interaction terms exist

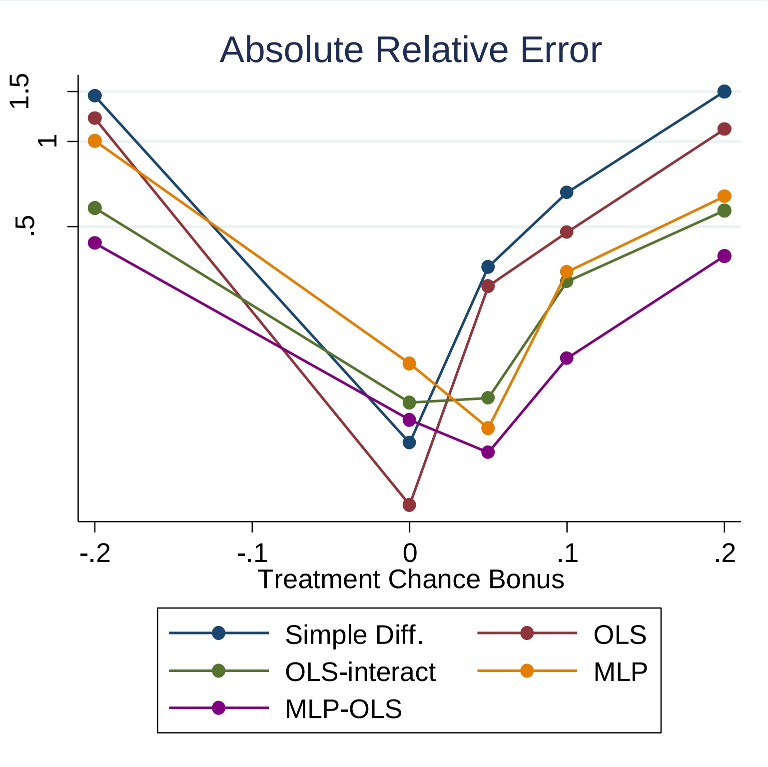


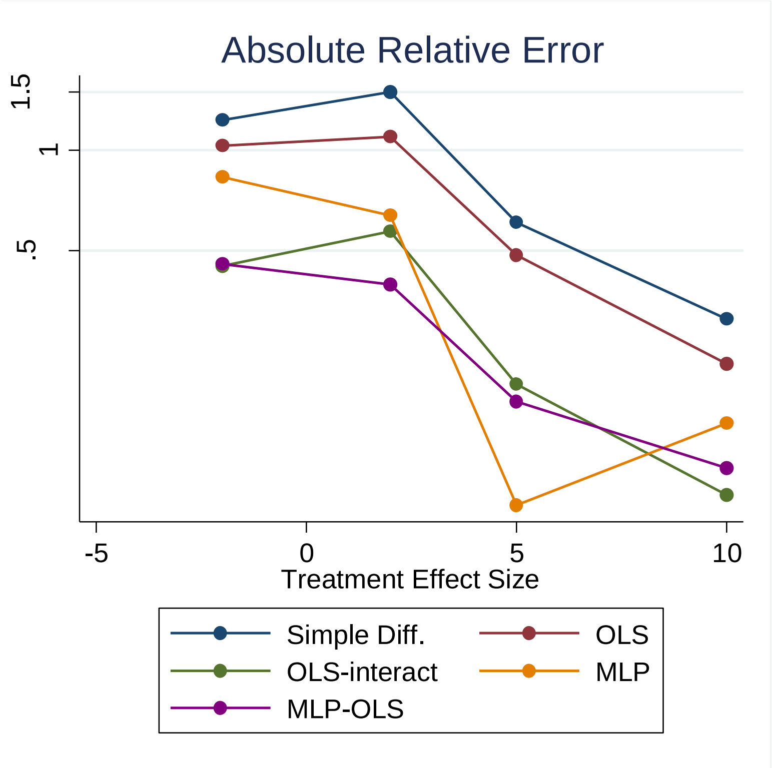


Settings: treatment effect=2, treatment chance bonus=0.2, single layer of 100 neurons

How Well Does Neural Network Perform?

- Across different treatment assignment bonus and treatment effects, MLP-OLS works better than OLS in most cases

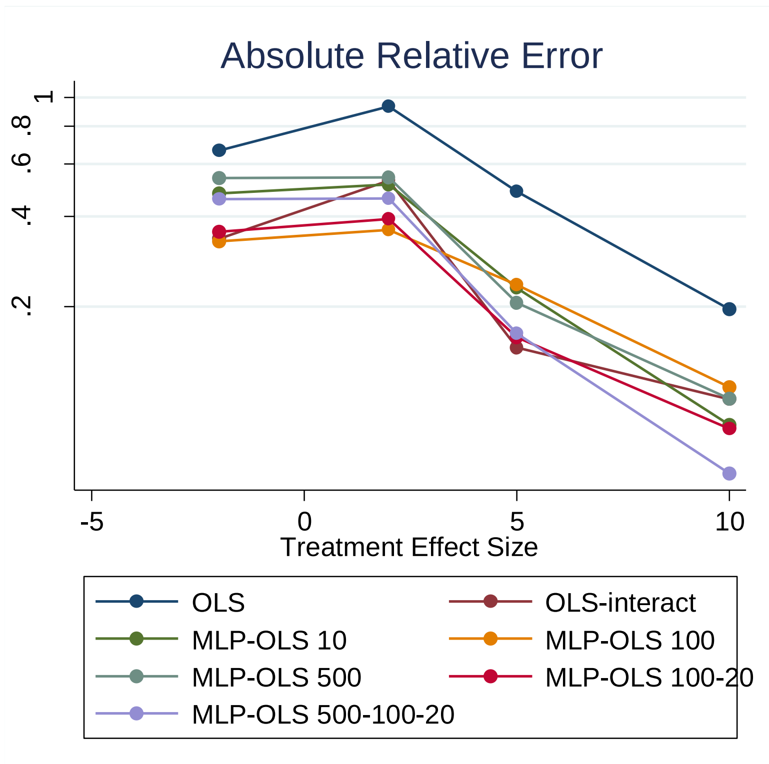


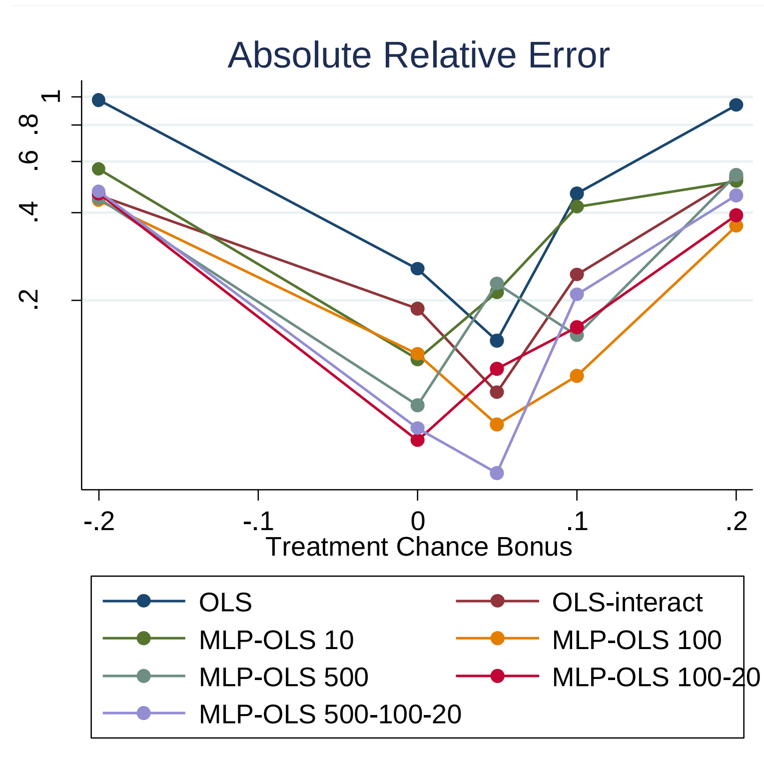


Settings: high order interactions, single layer of 100 neurons

Choosing the Right Parameters

- There is always a neural network that work better than OLS, but not necessarily the same one
- Cross-validation is necessary

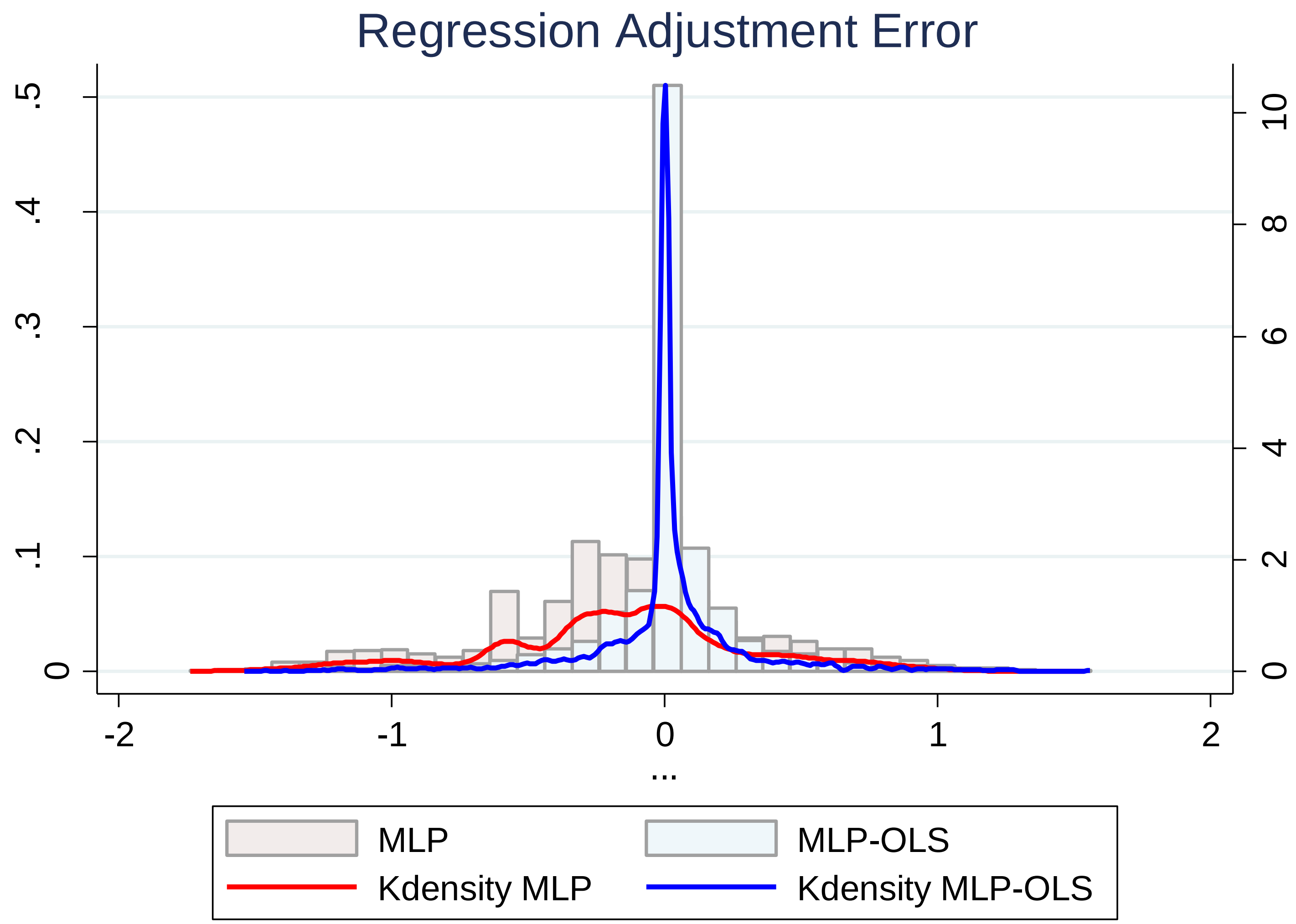




Settings: high order interactions

Distribution of Estimate Errors

- MLP-OLS appears unbiased while pure MLP does not



License Plate Data

- Recurrent neural network
  - 7 recurrent layers, 256 neurons per layer
  - Each character on license plate represented by 96 parameters