Predicting Enrollment at University of Iowa

Using Adversarially Debiased Neural Networks and Ensemble Trees

ISIB Research Symposium

Anthony Hobson, Aiden Orth, Carina Scholtens, Ryan Thoreson

Dr. Grant Brown

July 17, 2025

Objectives & Motivations

- **Goal:** Properly predicting enrollment during application cycle
- **Motivations**
  - Used by Administrators, Housing, and Academic Advisors
  - Preparation for future class years

Objectives & Motivations

- **Goal:** Properly predicting enrollment during application cycle
- **Motivations**
  - Used by Administrators, Housing, and Academic Advisors
  - Preparation for future class years
- **Objectives**
  - Explore neural networks in enrollment prediction
  - Explore possible directions of the current enrollment model

Objectives & Motivations

- **Goal:** Properly predicting enrollment during application cycle
- **Motivations**
  - Used by Administrators, Housing, and Academic Advisors
  - Preparation for future class years
- **Objectives**
  - Explore neural networks in enrollment prediction
  - Explore possible directions of the current enrollment model
- **Current Models**
  - Ensemble tree methods
    - Random Forests
    - Gradient Boosted Trees
  - Room for improvement

Data Overview

- Individual Data
  - Scholarship money
  - Distance from University of Iowa
  - Orientation Attendance
  - 90+ predictive variables
- Collected weekly during application cycles
- Six years of data
  - The tracked data varies by year
- Data Cleaning
  - Continuous: NAs set to 0; NA identification factor added
  - Categorical: NAs added as factor level

Data Used & Model Size

- Aligned Data
  - Data used from similar time in admission cycle in different years
  - Also uses 4 different years of data
  - Data presented is from middle of cycle
- Unaligned Data
  - Data collected from different periods in admission cycle
  - Data from 4 different years
- Model tested on data from years not in the training set

What is a Neural Network?

- Machine learning technique
- Models highly nonlinear relationships
  - Layers of linear and nonlinear functions
  - Sequential building of layers
- Increasing model complexity could lead to better predictions
- Many parameters to be learned
  - Very accurate results
  - Fear of overfitting
- Two model sizes
  - Small: 565,000 parameters
  - Large: 1,217,000 parameters

Neural Network Architecture

```{r}

net <- nn_module(
  initialize = function(d_in, d_hidden_1, d_hidden_2, d_out){
    self$net <- nn_sequential(
      nn_linear(d_in, d_hidden_1),
      nn_relu(),
      nn_linear(d_hidden_1, d_hidden_2),
      nn_relu(),
      nn_linear(d_hidden_2, d_out)
    )},
  forward = function(x){
    self$net(x)
  }
)


```

Figure 1: Neural Network using Torch [5]

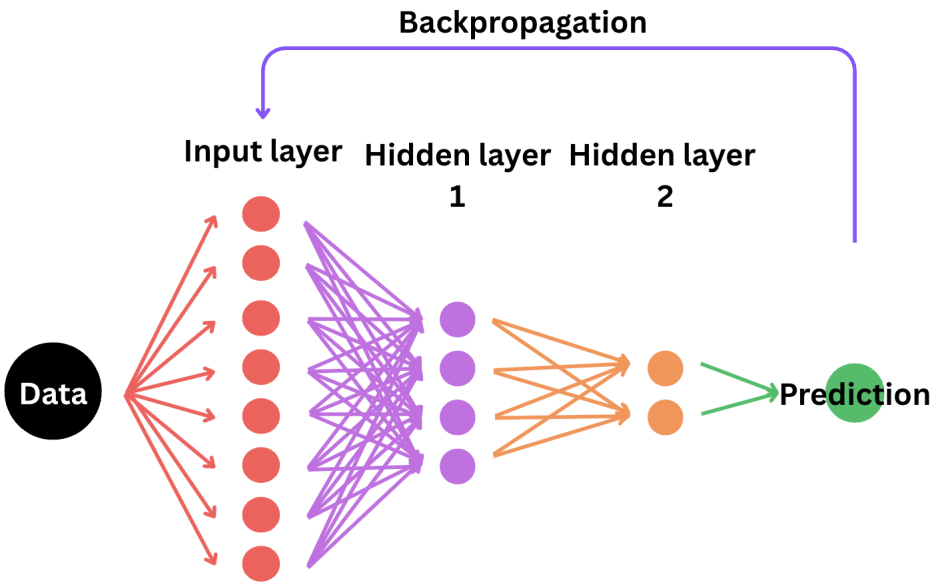


Figure 2: Network Architecture

Network Training

- Forward Pass
  Takes Inputs $\vec{X}$ and Predicts enrollment, Y

$$\hat{Y} = f(\vec{X}) = \sigma \circ T_3 \circ ReLU \circ ... \circ T_1(\vec{X})$$

Network Training

- Forward Pass
  Takes Inputs $\vec{X}$ and Predicts enrollment, Y

$$\hat{Y} = f(\vec{X}) = \sigma \circ T_3 \circ ReLU \circ ... \circ T_1(\vec{X})$$

- Loss Function
  Quantifies Accuracy

$$L(\hat{y}, y) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{2} \mathbb{I}\{y_i = k\} \cdot \log(P(y_i = k | x_i, \theta))$$

Network Training

- Forward Pass
  Takes Inputs $\vec{X}$ and Predicts enrollment, Y

$$\hat{Y} = f(\vec{X}) = \sigma \circ T_3 \circ ReLU \circ ... \circ T_1(\vec{X})$$

- Loss Function
  Quantifies Accuracy

$$L(\hat{y}, y) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{2} \mathbb{I}\{y_i = k\} \cdot \log(P(y_i = k | x_i, \theta))$$

- Backward Pass
  - Compute Gradients
  - Update Parameters
    - Stochastic Gradient Descent
    - ADAM Optimization Function
    - Variation of Learning Rate

Adversarial Neural Networks

- GAN - Generative Adversarial Networks [3]
  - Two networks: a generator creating simulated data and a discriminator trying to detect it
- Idea: One network, two predictions
  - Enrollment: Predict whether the student will enroll as accurately as possible
  - Year: Predict the year in which the student is applying as poorly as possible
- Input data strongly described by unimportant variable (Year)
- Two predictors are "Adversaries"
- Debiases Predictions [7]

Adversarially Debiased Architecture

```r
{r}
# Adversarial network
anet <- nn_module(
  initialize = function(d_in, d_hidden_1, d_hidden_2, d_out, num_classes_z){
    self$net <- nn_sequential(
      nn_linear(d_in, d_hidden_1),
      nn_relu(),
      nn_linear(d_hidden_1, d_hidden_2),
      nn_relu()
    )

    self$binary_head <- nn_sequential( nn_linear(d_hidden_2, d_out))
    self$multi_head <- nn_sequential( nn_linear(d_hidden_2, num_classes_z))
  },
  forward = function(x){
    features <- self$net(x)
    y_pred <- self$binary_head(features)
    z_pred <- self$multi_head(features)
    list(y_pred = y_pred, z_pred = z_pred)
  },

```

Figure 3: Adversarially Debiased Neural Net[5]

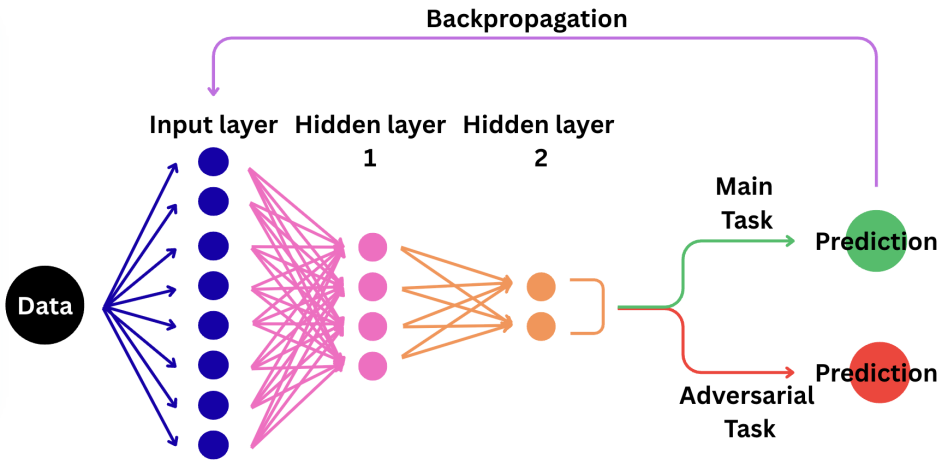


Figure 4: Network Architecture

Adversarially Debiased Loss Function

- Enrollment loss function: [4]

$$L_1(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{2}\mathbb{I}\{y_i = k\} \cdot \log(P(y_i = k|x_i, \theta))$$

Adversarially Debiased Loss Function

- Enrollment loss function: [4]

$$L_1(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{2}\mathbb{I}\{y_i = k\} \cdot \log(P(y_i = k|x_i, \theta))$$

- Year loss function:

$$L_2(\hat{z}, z) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{4}\mathbb{I}\{z_i = k\} \cdot \log(P(z_i = k|x_i, \theta))$$

Adversarially Debiased Loss Function

- Enrollment loss function: [4]

$$L_1(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{2}\mathbb{I}\{y_i = k\} \cdot \log(P(y_i = k|x_i, \theta))$$

- Year loss function:

$$L_2(\hat{z}, z) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{4}\mathbb{I}\{z_i = k\} \cdot \log(P(z_i = k|x_i, \theta))$$

- Adversarial loss function:

$$L(\hat{y}, \hat{z}, y, z) = L_1(\hat{y}, y) - \lambda \cdot L_2(\hat{z}, z)$$

Experiment Results

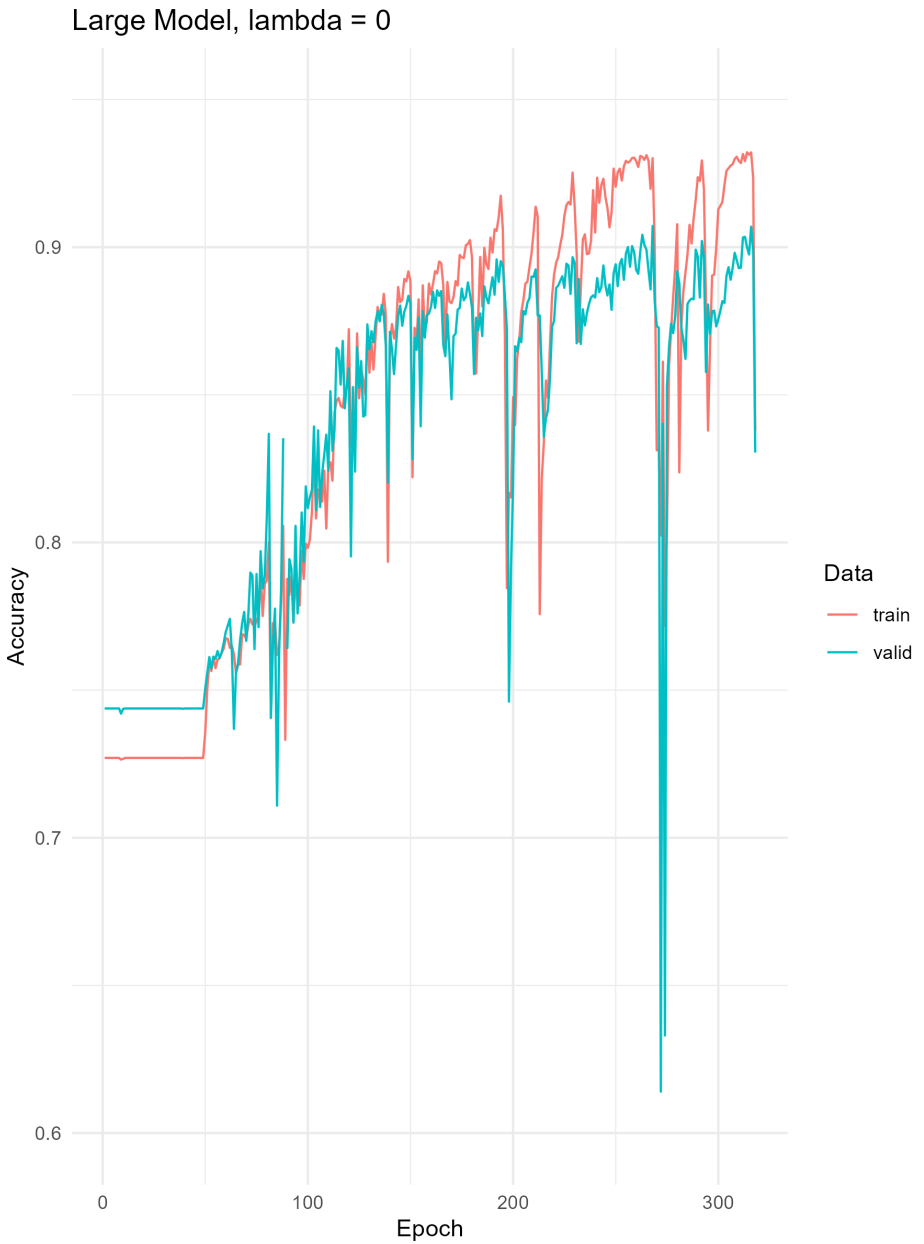
Model Size	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.5$
Small	0.869	0.809	0.846
Large	0.867	0.750	0.858

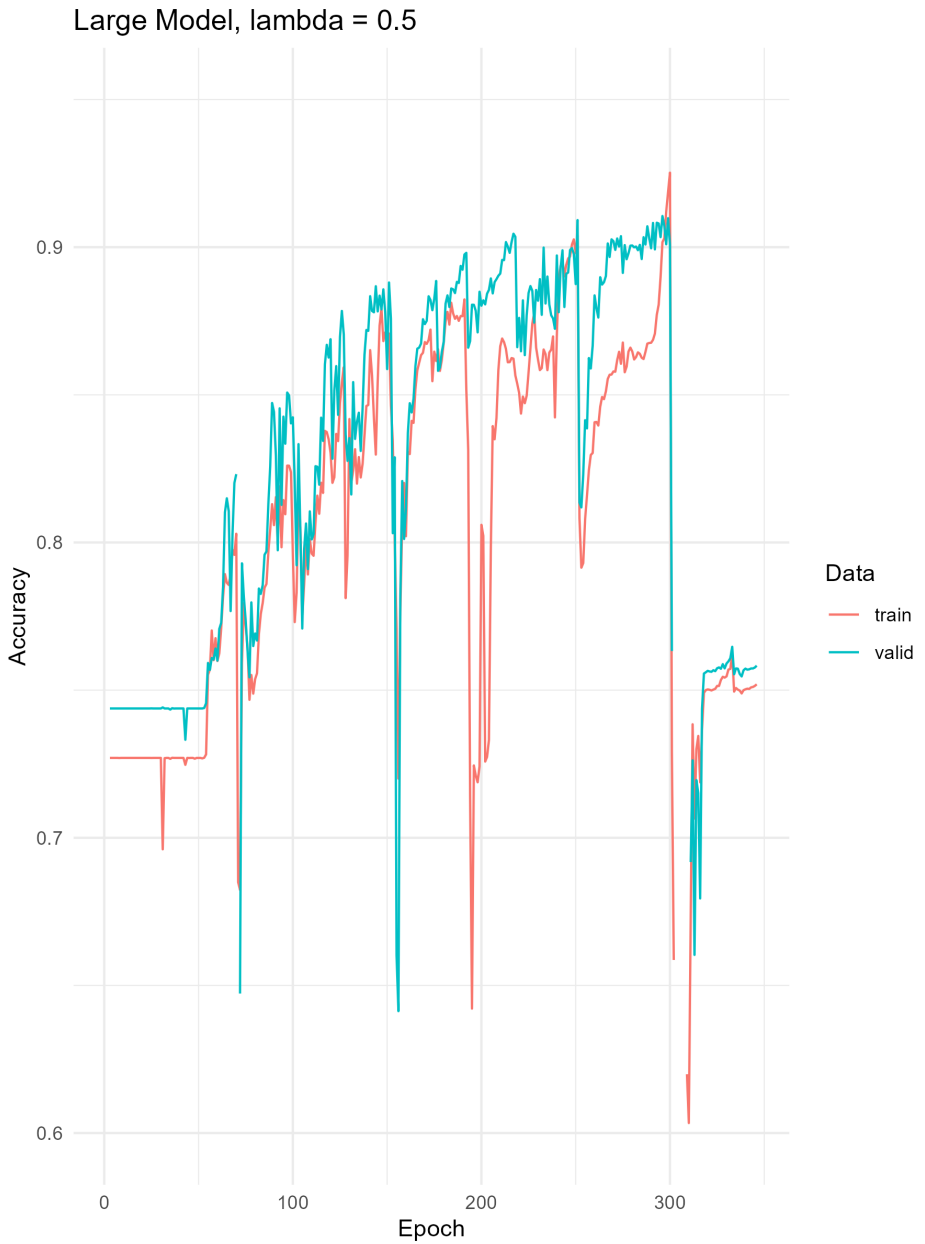
Table 1: Unaligned Data

Model Size	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.5$
Small	0.743	0.910	0.898
Large	0.907	0.740	0.911

Table 2: Aligned Data, Middle of Cycle

Comparison Plot





Random Forest Model

- Builds and combines multiple decision trees in training using bagging [6]
- Randomness introduced in the creation of new trees
- Outputs the majority vote for classification tasks

Random Forest Model

- Builds and combines multiple decision trees in training using bagging [6]
- Randomness introduced in the creation of new trees
- Outputs the majority vote for classification tasks
- Unaligned accuracy of: 0.864
- Aligned accuracy of: 0.922
- Run time of 1-2 minutes

XGBoost Model

- Makes predictions by combining outputs of many decision trees built sequentially [2]
- Each new tree is trained to correct the errors of the previous ones
- Uses gradient boosting to minimize a loss function based on residuals of the previous tree [1]

XGBoost Model

- Makes predictions by combining outputs of many decision trees built sequentially [2]
- Each new tree is trained to correct the errors of the previous ones
- Uses gradient boosting to minimize a loss function based on residuals of the previous tree [1]
- Unaligned accuracy of: 0.906
- Aligned accuracy of: 0.922
- Runtime: 10-15 seconds

Model Comparisons

	Neural Net.	XGBoost	Random Forest
Unaligned Data	0.867	0.906	0.864
Aligned Data	0.911	0.922	0.922

Table 3: Accuracy Model Comparison Aligned

	Neural Net.	XGBoost	Random Forest
Sensitivity	0.967	0.960	0.525
Specificity	0.808	0.712	0.981

Table 4: Unaligned Data: Sensitivity & Specificity

	Neural Net.	XGBoost	Random Forest
Sensitivity	0.962	0.980	0.753
Specificity	0.860	0.753	0.981

Table 5: Aligned Data: Sensitivity & Specificity

Overall Conclusions

- Should the University Admissions Cycle team adopt our model?
- Additional investment into an Adversarial Debiasing Network may prove to be successful but cumbersome
- Inconsistency in the training process makes a current commitment to neural networks impractical
- While Neural Networks may not be the best way to predict enrollment, adversarial debiasing is a promising direction to continue exploring

References I

[1] Leo Breiman. "Random forests.". In: *Machine learning* 45.1 (2001), pp. 5–32.

[2] Tianqi Chen et al. *xgboost: Extreme Gradient Boosting*. R package version 1.7.7.1. 2024. URL: `https://CRAN.R-project.org/package=xgboost`.

[3] Antonia Creswell et al. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

[4] Daniel Falbel and Javier Luraschi. *torch: Tensors and Neural Networks with 'GPU' Acceleration*. R package version 0.14.2. 2025. URL: `https://CRAN.R-project.org/package=torch`.

[5] Sigrid Keydana. *Deep Learning and Scientific Computing with R torch*. Chapman and Hall/CRC, 2023.

References II

[6] Marvin N. Wright and Andreas Ziegler. “ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R”. In: *Journal of Statistical Software* 77.1 (2017), pp. 1–17. DOI: 10.18637/jss.v077.i01.

[7] Han Xu et al. “To be robust or to be fair: Towards fairness in adversarial training”. In: *Proceedings of Machine Learning Research* (2021).

Thank You!

ISIB Program sponsored by the National Heart Lung and Blood Institute

Grant HL161716-01.

Dr. Grant Brown

Iowa Department of Biostatistics