Title: Using Deep Learning to Measure Effect Sizes of Isolated Variables by Eliminating Confounders

Author: Joseph Zhu

Email: josephzhu.academic@gmail.com

Affiliation: University of New South Wales, School of Clinical Medicine and School of Computer Science

Abstract: Current methods of calculating effect sizes of variables are unable to evaluate them precisely due to unavoidable confounding biases despite best attempts to account for them. This paper proposes the use of deep learning in neural networks (NN) to eliminate the effects of measured confounders by simulating an environment where researchers can directly compare the effect sizes of a variable on otherwise identical groups of participants. It also demonstrates a proof-of-concept for the feasibility of this method using a fully connected feedforward network to evaluate the RR of mortality in using D-penicillamine compared to placebo in the treatment of advanced PBC based on the Mayo Clinic PBC dataset. A secondary outcome of the RR of mortality in females compared to males after adjusting for D-penicillamine and placebo was also examined. Predictions of effect sizes of both primary and secondary outcomes are comparable to currently established values. The use of deep learning in neural networks proves promising in improving the calculations of effect size of an isolated variable through eliminating measured confounding factors. This method may be even more effective in analysing systematic reviews, case-control, or large-scale studies due to their large dataset size and beneficial study designs.

# Table of Contents

[**Table of Contents 2**](#_yug45eeqfmck)

[**Introduction 3**](#_ieq0uz7yr45m)

[**Method 4**](#_t0ossxd3ti71)

[Data Gathering and Preprocessing 4](#_qchupfoy2ar5)

[Model Selection and Training 4](#_mtlueea7pxti)

[Model Specifications 4](#_8ibyegszswko)

[Input Data Formatting 5](#_k7s4s8jgm01y)

[Determining the Optimal Model Structure 5](#_91ys1khc37pm)

[**Results 7**](#_3wrjun2hirrm)

[**Discussion 8**](#_6222dp39cg2x)

[Interpretations of Results 8](#_3tnucfk5g3o2)

[Main Outcome 8](#_2pncea6l4j2e)

[Secondary Outcome 8](#_f7ayfzc1db0b)

[Limitations 9](#_fl6b0eauhyah)

[Possible Uses 9](#_7x94uad42bun)

[**Conclusion 11**](#_7tr2zm9e8bim)

[**References: 12**](#_c5ypv2ublpma)

[**Appendix A: Variable Labels in the PBC Dataset 13**](#_12c3lfja9o2z)

[**Table 1 14**](#_f5ulxh3l9pjm)

[**Table 2 15**](#_poluun4i4p15)

[**Table 3 16**](#_pygwtt2d82qh)

# Introduction

Effect sizes of variables, often denoted by relative risk (RR) and odds ratio (OR) in studies, are the most important measure in empirical studies.1 However, current methods of calculating it are unable to evaluate it precisely due to myriads of confounding factors. Although researchers may attempt to eliminate these confounders by sampling participants with similar health, risk factors, and history, it is impossible to account for all factors present due to unavoidable variations between individuals.

Ideally, one would compare the effects of the intervention to control on the same individual; however, this is impossible in the real world as no two people are exactly alike. This paper proposes the use of deep learning in neural networks (NN) to eliminate the effects of measured confounders by simulating an environment where researchers can directly compare the effect sizes of a variable, such as drug vs. control in the previous example or a gene mutation on the rates of cancer, on otherwise identical groups of participants.

Given a dataset and target outcomes, it is possible to train a NN to predict the outcomes to some degree of accuracy. If the accuracy is high, the model can reliably predict outcomes given a set of inputs. By duplicating the dataset and setting all inputs of a variable of one dataset to control and the other dataset to intervention, the mode can estimate the exact relative risk without confounding biases from other variables. Furthermore, statistical analysis of the predictions across a wide range of models generated from this process can improve confidence in the estimation of true effect size.

The following section describes a demonstration of an application of the aforementioned process. However, the aim of this paper is not to provide a guide on how to create the optimal NN, but rather to provide a proof-of-concept for the efficacy of using deep learning in the estimation the true effect sizes of isolated variables by eliminating confounders, as well as to examine limitations and possible uses of this method.

All generating code, as well as statistics for each model trained, are publicly available at https://github.com/jzhu-academic/deep-learning-to-measure-effect-sizes.

# Method

## Data Gathering and Preprocessing

The dataset used for this proof-of-concept is from the Mayo Clinic trial in primary biliary cholangitis (PBC), known at the time as primary biliary cirrhosis, conducted between 1974 and 1984.2,3 During preprocessing, the 106 cases where the participant did not participate in the clinical trial were removed with 312 valid cases remaining. The variables measured and their representation in the dataset as described by Mayo Clinic can be found in Appendix A. Missing elements in the dataset were indicated by a period.

The NN’s prediction is “Status”, denoting the outcome of the patient. The primary outcome examined is the RR of mortality between using D-penicillamine compared to placebo in the treatment of advanced PBC. A secondary outcome of the RR of mortality in males compared to females after adjusting for D-penicillamine and placebo was also examined.

## Model Selection and Training

### Model Specifications

The model is a fully connected feedforward network coded in Python 3.12.4, with the core of the NN built using the PyTorch4 library. To represent the NN structure, [A:B] is defined to represent such a model with *A* neurons in the first hidden layer, and *B* neurons in the second. The pre-output layer then maximally connects to the three output neurons which represent the outcomes “alive”, “liver transplant”, and “dead”. A Leaky Rectified Linear Unit (LeakyReLU) is applied to each hidden layer as its non-linear activation function. As this is a multi-class classification problem, each NN uses a cross-entropy loss function; it also uses stochastic gradient descent (SGD) with momentum as its optimiser (Figure 1). The hyperparameters were arbitrarily set to 0.01 and 0.1 for the learning rate and momentum of the optimiser, and to 0.3 for the gradient for negative inputs to LeakyReLU initially.

Every NN generated is trained 200 times, known as epochs, on the training data, which represents 80% of cases of the dataset. Each epoch is trained on a random permutation of the training set with the inputs being fed in by batches of 32. A majority vote by five of such NNs (referred to collectively as a tree), which are all trained on identical training data, determines the final outcome prediction; this is to improve prediction confidence by accounting for natural variation that occurs during training.

This process is repeated 200 times (referred to as spans to differentiate from epochs) for each set of hyperparameters, with each span generating a random split of the dataset into training and testing data. To calculate the predicted RR of mortality between D-penicillamine and placebo, the span duplicates the dataset, before setting all entries of “Drug” to D-penicillamine in one dataset and setting all entries placebo in the other (Figure 2A). The model can then predict the mortality rate of both datasets and divide that of D-penicillamine to that of the placebo to get the predicted true RR after eliminating confounding variables (Figure 2B). Each span then records the predicted RR, as well as the accuracy of its tree calculated based on the remaining 20% of unseen testing data.

The secondary outcome of male to female mortality RR is calculated by first setting all inputs for “Drug” to placebo, then following the same steps as previously outlined to create two copies of the dataset, one with all sex as males and the other as all females.

The key comparators used to determine the performance of NN structures are its prediction accuracy on the testing data, and the variance in its spans’ predictions of the true RR of mortality as a measurement of the model’s confidence.

### Input Data Formatting

Missing values for numerical labels (i.e. Age, Bilirubin, Cholesterol, Albumin, Copper, Alk\_Phos, SGOT, Triglycerides, Platelets, Prothrombin) in the dataset were filled in with the median of that column. The numerical labels were then formatted as a proportion of the maximum value of that column to squeeze all values between 0 and 1. Values for the non-ordinal labels “Sex” and “Drug” were reformatted using one-hot encoding to prevent the introduction of bias associated with ordinality.

No significant difference (p<0.01) in accuracy was found between one-hot encoding the categorical labels “Ascites”, “Hepatomegaly”, “Spiders”, “Edema”, and “Stage” nominally compared to keeping it as an ordinal input (Table 1) in both NNs with low and high parameter counts. This is likely due to the ordinal label acting as a measurement of the severity of symptoms. There was also no observable benefit in variance of one labelling method over another. Thus, these variables were stored ordinally to reduce the number of inputs and minimise training time.

### Determining the Optimal Model Structure

The results from Table 1 suggest a benefit in using NNs with one hidden layer compared to those with more for this dataset. To confirm this, NNs with similar parameter counts but different numbers of layers are compared against each other (Appendix A). Across all parameter counts, those with one hidden layer had higher accuracy (Figure 3A) and lower variance (Figure 3B) compared to those with more layers but similar parameter counts. In the NNs with one hidden layer, predictive accuracy increased and predicted RR variance decreased up to [1536]. Further increases in the layer size did not yield improvements in accuracy or variance, indicating possible overfitting.

To ensure that the hidden layer was beneficial at all, the result was compared to a neural network with no hidden layers, as denoted by [ ]. This demonstrated worsened variance and accuracy, indicating the advantage of the nonlinearity introduced by the hidden layer (Table 2).

To fine-tune the epochs, learning rate (LR), and momentum (M), predictions by a [1536] with different sets of hyperparameters were compared (Table 3). The best hyperparameter set was determined by a score with 50% weighting on its mean prediction accuracy proportional to the range of all model accuracies and 50% weighting on its predicted RR variance proportional to the range of all predicted RR variances after excluding the significant (p<0.05) outlier, [400 epochs, 0.02LR, 0.6M]. Following this scoring method, the best hyperparameter set for [1536] was determined to be [200 epochs, 0.01LR, 0.3M]. The model [1536] with hyperparameter set [200 epochs, 0.01LR, 0.3M] was then used to predict the RR of mortality of D-penicillamine compared to placebo in treating PBC over 1000 spans.

# Results

This model predicted a mortality RR of 1.0864 (95% CI 1.0823-1.0905) in taking D-penicillamine compared to placebo in patients with PBC, with a prediction accuracy of 0.7712 (95% CI 0.7681-0.7743).

The secondary outcome of RR of mortality in males compared to females was also measured using the same model structure over 1000 spans and predicted a RR of 1.5642 (95% CI 1.5490-1.5795).

# Discussion

## Interpretations of Results

### Main Outcome

Across the 1000 spans, the predicted true RR associated solely with switching from placebo to D-penicillamine in the treatment of PBC is 1.0864 (95% CI 1.0823-1.0905). This result differed from the RR of 1.34 (95% CI 1.09-1.64) found by Gong et al. in their systematic review,5 but overlapped with its confidence interval and agreed with the conclusion that D-penicillamine increases the risk of mortality compared to placebo in the treatment of PBC. However, this does not mean that the difference between the two results is insignificant6; without knowing the details of the data collected by John et al., it is impossible to determine whether there exists a statistically significant difference. If the difference is significant, there are several possibilities that may have caused this.

Regarding this proof-of-concept, the model was simple and shallow, the dataset size was limited, and the accuracy was not optimal. Furthermore, all models trained in this paper failed to predict the outcome of “liver transplant” correctly due to its rarity in the dataset, instead preferring to label cases as either “alive” or “dead” whenever the true outcome was “liver transplant”. The model’s accuracy could likely be improved upon by utilising techniques such as dropout and other regularisation methods to support a deeper NN to learn more complex relationships without causing overfitting; the accuracy may also be improved by integrating techniques such as random forests or bootstrapping to account for the small dataset size and improve predictions of rare outcomes.

Regarding the systematic review, the trials reviewed were highly heterogeneous, especially in regard to mortality.4 The inconsistencies between trials also extended to their participant sampling, thereby introducing confounding factors. Since the aim of this process was to eliminate the confounding biases present, it is expected to find a different RR, especially in a systematic review with greatly heterogeneous trials such as this.

### Secondary Outcome

The RR of mortality in males compared to females with PBC is much more commonly studied and is well-established. The NN model calculated the mortality RR to be 1.5642 (95% CI 1.5490-1.5795); comparatively, the adjusted hazard ratio of males to females with PBC determined to be 1.80 (95% CI 1.01-3.19) by John et al.7 These results do not significantly differ (p<0.05), further serving to demonstrate the feasibility of using NNs to determine effect size.

## Limitations

This proposed method of using NNs to estimate the true RR suffers from many of the same problems plaguing NNs across all fields. It requires numerical or ordinal inputs, or for categorical data to be encoded by a process such as one-hot encoding. In datasets with categorical labels with high cardinality, this method of encoding will inflate model size which can complicate the model. One method of overcoming this problem is by grouping rare inputs into one larger classification. This method is also impractical in studies with a low number of participants. Without sufficient data, it is difficult to accurately and reliably train a NN to predict the outcomes, especially when a portion of the data must be sectioned off for testing. Furthermore, the presence of rare outcomes, such as “liver transplant” in this proof-of-concept, hinders the models’ accuracy.

It should be noted that this method, like any other empirical study, relies on a sample that is representative of the population. The reliability of the NN model can only calculate intervariable relationships to the extent of the dataset’s accuracy. Similarly, the NN is only capable of removing confounders that were measured in the collection of data. Researchers should not solely rely on this method in eliminating biases, but continue to account for possible sources of error.

## Possible Uses

The method of assessing effect sizes proposed in this paper can also be extended to studies with multiple outcomes such as assessing the RR of side effects occurring. So long as the inputs and outputs are separated prior to the training of models, NNs can be used to predict the effect sizes for each outcome independently. Furthermore, a low predictive accuracy, or a difference in effect size calculated by current methods and by NNs, can also indicate to the researchers of unmeasured or uncontrolled variables contributing to the outcomes.

The design of the study can be used to overcome many of the limitations this method faces:

* Systematic reviews can utilise this method since grouping the independent trials provides a larger dataset for training and testing.
* Case-control studies effectively eliminate the problem of rare outcomes by sampling the participants with and without the outcome being studied proportionally.
* Using NNs to isolate the effect of risk factors also improves the efficiency of large-scale prospective and cross-sectional studies. Risk factors and their effect size can be isolated and intervariable relations that were previously unable to be understood can now be deduced using this process.

This method may also be used to identify previously unknown confounding factors and assess confounding strength by individually assessing each measured variable.

# 

# Conclusion

The use of deep learning in neural networks proves promising in improving the calculations of effect size of an isolated variable through eliminating measured confounding factors. Despite the simplicity of the model presented in this paper, it demonstrated comparable results to currently established values. This method may be even more effective in analysing systematic reviews, case-control, or large-scale studies due to their large dataset size and beneficial study designs. The use of AI to eliminate confounders may also be applied to evaluate intervariable interactions and warn researchers of unmeasured or uncontrolled variables, as well as identify new confounding factors. With this method, future studies are encouraged to collect more variables in order to process and eliminate a greater proportion of confounders.

# References:

1. Lakens D. Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs. Frontiers in Psychology. 2013;4(863). doi:https://doi.org/10.3389/fpsyg.2013.00863

2. Fleming TR, Harrington DP. Appendix D: Data. Wiley series in probability and statistics. Published online September 8, 2005:359-383. doi:https://doi.org/10.1002/9781118150672.app4

3. Mayo Clinic. Date unknown. Accessed October 14, 2024. https://www.mayo.edu/research/documents/pbchtml/doc-10027635

4. Ansel J, Yang E, He H, et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. Published online April 27, 2024. doi:https://doi.org/10.1145/3620665.3640366

5. Gong Y, Klingenberg SL, Gluud C. D-penicillamine for primary biliary cirrhosis. *Cochrane Database of Systematic Reviews*. Published online October 18, 2004. doi:https://doi.org/10.1002/14651858.cd004789.pub2

6. Greenland S, Senn SJ, Rothman KJ, et al. Statistical tests, P values, Confidence intervals, and power: a Guide to Misinterpretations. European Journal of Epidemiology. 2016;31(4):337-350. doi:https://doi.org/10.1007/s10654-016-0149-3

7. John B, Aitcheson G, Schwartz K, et al. Male Sex Is Associated With Higher Rates of Liver‐Related Mortality in Primary Biliary Cholangitis and Cirrhosis. *Hepatology*. 2021;74(2):879-891. doi:https://doi.org/10.1002/hep.31776

# Appendix A: Variable Labels in the PBC Dataset

* Number of days between registration and the earlier of death, transplantation, or study analysis time in July, 1986
* Status: 0=alive; 1=liver transplant; 2=dead
* Drug: 1=D-penicillamine; 2=placebo
* Age in days
* Sex: 0=male; 1=female
* Presence of ascites: 0=no; 1=yes
* Presence of hepatomegaly 0=no; 1=yes
* Presence of spiders 0=no; 1=yes
* Presence of edema 0=no edema and no diuretic therapy for edema; 0.5 = edema present without diuretics, or edema resolved by diuretics; 1 = edema despite diuretic therapy
* Serum bilirubin in mg/dl
* Serum cholesterol in mg/dl
* Albumin in gm/dl
* Urine copper in ug/day
* Alkaline phosphatase in U/litre
* SGOT in U/ml
* Triglycerides in mg/dl
* Platelets per cubic ml / 1000
* Prothrombin time in seconds
* Histologic stage of disease

# Table 1

Table 1. Comparison of nominal and ordinal labelling across different NNs with similar parameter counts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nominal (one-hot) labelling | | Ordinal (numerical) labelling | |
| NN Size | Pred. RR Variance | Accuracy (95% CI) | Pred. RR Variance | Accuracy (95% CI) |
| [1024]  24579 parameters | 0.0065 | 0.7652 (0.7585-0.7720) | 0.0054 | 0.7678 (0.7609-0.7747) |
| [144:144]  24339 parameters | 0.0066 | 0.7557 (0.7448-0.7626) | 0.0081 | 0.7428 (0.7356-0.7499) |
| [104:104:104]  24339 parameters | 0.0090 | 0.7531 (0.7466-0.7596) | 0.0090 | 0.7486 (0.7415-0.7557) |
| [88:88:88:88]  25611 parameters | 0.0201 | 0.7048 (0.6976-0.7121) | 0.0369 | 0.7174 (0.7105-0.7242) |
| [1792]  43011 parameters | 0.0046 | 0.7663 (0.7595-0.7731) | 0.0045 | 0.7695 (0.7625-0.7756) |
| [192:192]  41667 parameters | 0.0075 | 0.7565 (0.7499-0.7631) | 0.0065 | 0.7578 (0.7512-0.7644) |
| [144:144:144]  45219 parameters | 0.0075 | 0.7509 (0.7440-0.7578) | 0.0079 | 0.7551 (0.7483-0.7619) |
| [120:120:120:120]  46443 parameters | 0.0188 | 0.7234 (0.7162-0.7307) | 0.0135 | 0.7299 (0.7229-0.7370) |

### 

# Table 2

Table 2. Comparison of [ ] and [1536]

|  |  |  |
| --- | --- | --- |
| NN Size | Predicted RR Variance | Accuracy (95% CI) |
| [ ] - 63 parameters | 0.0145 | 0.7156 (0.7080-0.7232) |
| [1536] - 36867 parameters | 0.0049 | 0.7727 (0.7660-0.7794) |

# Table 3

Table 3. Comparison of different hyperparameter sets

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 100 Epochs | | 200 Epochs | | 400 Epochs | |
|  | Pred. RR Variance | Accuracy  (95% CI) | Pred. RR Variance | Accuracy  (95% CI) | Pred. RR Variance | Accuracy  (95% CI) |
| 0.005 LR  0.05 M | 0.0080 | 0.7511 (0.7443-0.7579) | 0.0061 | 0.7615 (0.7544-0.7686) | 0.0048 | 0.7674 (0.7606-0.7742) |
| 0.005 LR  0.1 M | 0.0094 | 0.7444 (0.7372-0.7515) | 0.0068 | 0.7663 (0.7599-0.7727) | 0.0043 | 0.7706 (0.7638-0.7773) |
| 0.005 LR  0.3 M | 0.0086 | 0.7538 (0.7468-0.7538) | 0.0043 | 0.7713 (0.7644-0.7783) | 0.0040 | 0.7714 (0.7648-0.7781) |
| 0.005 LR  0.6 M | 0.0058 | 0.7626 (0.7557-0.7695) | 0.0040 | 0.7694 (0.7628-0.7759) | 0.0046 | 0.7747 (0.7684-0.7809) |
| 0.01 LR  0.05 M | 0.0060 | 0.7625 (0.7553-0.7697) | 0.0049 | 0.7668 (0.7596-0.7740) | 0.0043 | 0.7691 (0.7623-0.7760) |
| 0.01 LR  0.1 M | 0.0052 | 0.7688 (0.7622-0.7754) | 0.0049 | 0.7727 (0.7660-0.7794) | 0.0042 | 0.7699 (0.7634-0.7765) |
| 0.01 LR  0.3 M | 0.0051 | 0.7694 (0.7629-0.7759) | 0.0041 | 0.7720 (0.7655-0.7785) | 0.0053 | 0.7606 (0.7543-0.7670) |
| 0.01 LR  0.6 M | 0.0046 | 0.7729 (0.7665-0.7792) | 0.0042 | 0.7671 (0.7603-0.7738) | 0.0066 | 0.7564 (0.7496-0.7632) |
| 0.02 LR  0.05 M | 0.0052 | 0.7660 (0.7586-0.7733) | 0.0047 | 0.7629 (0.7565-0.7692) | 0.0067 | 0.7590 (0.7524-0.7656) |
| 0.02 LR  0.1 M | 0.0038 | 0.7601 (0.7530-0.7672) | 0.0052 | 0.7663 (0.7597-0.7730) | 0.0069 | 0.7509 (0.7437-0.7581) |
| 0.02 LR  0.3 M | 0.0048 | 0.7629 (0.7558-0.7701) | 0.0056 | 0.7627 (0.7560-0.7694) | 0.0061 | 0.7473 (0.7407-0.7539) |
| 0.02 LR  0.6 M | 0.0057 | 0.7587 (0.7517-0.7658) | 0.0072 | 0.7586 (0.7507-0.7665) | 0.0109 | 0.7396 (0.7329-0.7464) |