Cognitive Bias Recognition of Trained Neural Networks

Hanna Fields

Advisor: Douglas Blank

A project presented for the degree of Bachelor of Arts in Computer Science

May 2017
Department of Computer Science
Bryn Mawr College
Bryn Mawr, Pennsylvania, US

Abstract

In this project, I examined techniques for quantifying cognitive bias from the results of neural networks. Cognitive bias refers to prejudiced judgments and inferences humans make that are rooted in experience. These inferences are not based on logic but false corollaries. False corollaries refer to factors that don’t influence an outcome, rather the outcome informs the factors. The two techniques for quantifying cognitive bias produced from this project are identifying proxy variables and measuring equality. Measuring equality is most applicable to neural networks for which there is a desired even distribution of results. The measuring equality technique can be used to combat selective bias, which arises when a network is not trained on a diverse dataset. The second technique is identifying proxy variables, which combats variable bias. Variable bias is the inclusion of variables in the training set for a neural network that can contribute to undesired bias in the results from the network. Both the identifying proxy variables and measuring equality techniques were implemented using Northpointe’s recidivism data to train a neural network. This data was the results of Northpointe’s neural network that predicts the likelihood of criminal recidivism. The results from the proxy variable technique were somewhat inconclusive, while the implementation of measuring equality showed that there was clear racial bias in the network. These results imply that there are possible techniques for quantifying bias in neural networks.

1. **Introduction**

Cognitive bias is a physiological term that is defined by the idea of bias rooted in experience. It refers to prejudiced judgments and inferences humans make that are not logical based on certain factors, but false corollaries (Cherry, 2017). False corollaries refer to factors that don't have an influence on an outcome, but potentially have an influence skewing a system's outcomes. The connection between the outcome and these factors is not bidirectional rather the outcome only informs the factors, and the factors by in large do not inform the outcome.

Training neural networks with biased datasets results in a network that imitates human cognitive bias. The outputs of neural networks are based on data that can reflect bias against historically marginalized groups. The bias reflected in such data is systemic in nature, and helps to reproduce racism, classism, and sexism when used to train neural networks.

The purpose of this project is to examine techniques for quantifying cognitive bias. The problems in this paper are addressed by techniques to quantify cognitive bias resulting from variable and selective bias. The two main issues which propagate biased networks generally stem from either having selectively biased data, or from not using the appropriate variables to weigh the decisions of neural network. The two techniques for quantifying cognitive bias produced from this project are identifying proxy variables and measuring equality.

Measuring equality is most applicable to neural networks for which there is a desired even distribution of results. The measuring equality technique can be used to combat selective bias, which arises when a network is not trained on a diverse dataset. The second technique is identifying proxy variables, which combats variable bias. Variable bias is the inclusion of variables in the training set for a neural network that can contribute to undesired bias in the results from the network.

2.1 Neural Networks

The terms that are relevant to this discourse of neural networks are described below. The training dataset is used to teach the network the problem that is to be solved, while the test dataset is used to generate results from the network. Inputs and outputs could refer to test or training set inputs and outputs. In general, inputs are variables that are fed to a network, for example time, race, and gender could be inputs. Outputs are a bit more nuanced; outputs of a training dataset refer to outputs that are then compared to the desired outputs or target. The correctness of which is then used to reweight, or "teach" the network. Outputs of a test dataset refers to outputs of a network that has already been trained, ideally these outputs match outputs from the test dataset if it is available. Correctness refers to the percentage of training results from a network that correctly matches the target value. Tolerance defines the range within which a result can be considered correct, and can thus contribute to the correctness percentage. Propagation is the feeding of a test input into a network and lastly, an epoch is a single run of a test dataset on a learning algorithm.

The neural network used in this project is a feedforward backpropagation of error network, otherwise known as a backpropagation network (Rumelhart, 1986). Neural networks are characterized by their lack of definition based around symbols and variables, and are instead defined by how nodes are connected and identify patterns. The feedforward part refers to the idea that the nodes of the network do not cycle between layers of the network, rather they feedforward their values from the input layer, hidden layers, and finally the output layer (Leverington, 2015). Lastly, the backpropagation piece refers to the backpropagation of the outputs' error through the network during each training trail.

2.2 Background and Related Work

Selectively biased datasets are those that do not contain enough diverse data, resulting in an absence of information for the neural network to base its conclusions on. An example of selective bias is Nikon and Hewlett-Packard's poor camera facial recognition of people of color. Specifically, Nikon's cameras had issues with sending notifications that people of Asian descent were blinking when they were not, and Hewlett-Packard's cameras had issues identifying the faces of people with dark skin tones; while it had no trouble identifying the faces of people with fairer skin tones (Rose, 2010). These facial recognition issues most likely stemmed from Nikon and Hewlett-Packard predominantly training their cameras to recognize white faces, and thus using a limited dataset.

In contrast to selective bias, an example of variable bias is Amazon's same-day delivery service being unavailable in historically redlined and predominantly black neighborhoods (Crawford, 2016). Since the negative press was released regarding this issue and the resulting backlash, Amazon has expanded their same-day delivery to cover all zip codes in Boston, Chicago, and New York (Ingold, 2016). The origin of Amazon's bias is less definitive than the camera recognition, since Amazon has not publicly divulged their algorithm for deciding same-day delivery service neighborhoods. However, Amazon has publically stated that it doesn't know the race of its customers (Ingold, 2016). Amazon's stated intention was to provide this service in areas with high concentrations of Amazon Prime members, near Amazon's product warehouses, and where mail carriers deliver up until 9 p.m. Customers who fit this description tend to live in middle class white neighborhoods (Cox, 2016). Furthermore, distance could not have been a highly-weighted variable since there were cases of certain neighborhoods being excluded that were surrounded by neighborhoods where the service was provided. Thus, it is apparent that Amazon did not base its conclusions for where to provide same-day delivery on the appropriate variables, but proxy variables which reflect systemic bias.

The Amazon and camera facial recognition examples had either missing variables that pertained to the problem being solved which were not being considered, or made conclusions based on the wrong variables. The key difference between selective and variable bias is that for selective bias the solution would be to collect more data, while in the case of variable bias the solution would be to use different variables.

2.3 Case Study - Word Association

An example of biased artificial intelligence resulting from prejudiced data is Princeton University’s AI version of the Implicit Association Test (IAT) (Caliskan-Islam, 2016). The intention of IAT in phycology is to test a subject’s associations with a given word. Princeton’s implementation of an IAT was based on the GloVe algorithm, a neural network based approach which was used to identify word embedding. Word embedding is pattern matching which words tend to appear close or next to each other, thereby identifying in which context the word appears. Within the GLoVe algorithm the researchers developed the Word Embedding Association Test (WEAT) to quantify bias, and the Word Embedding Factual Association Test (WEFAT) in order to ensure words with multiple meanings were analyzed correctly (Caliskan-Islam, 2016). An example of the use of WEFAT would be accounting for the fact that the name “Will” is also a proper word, WEFAT would make a judgement based on the context that “Will” appears to determine if it “name like” enough to be included in IAT test. The results of running this algorithm on data from twitter found that white names were more likely to be associated with positive words and black names were more likely to be associated with negative words in text. Another stereotypical display of bias that was found was that female names tend to be associated with words related to ‘family’, and male names with words related to ‘career’ (Collins, 2016). Princeton’s IAT implementation is a good example of how biases are learned from context for humans and artificial intelligence.

2.4 Case Study – Governor Neural Networks

Governor neural networks are networks that were developed to create networks that learn better (Blank, 2004). Governor neural networks are an example of an artificial intelligence algorithm that can combat issues that arise when a dataset lacks diversity within the categories it considers. Lacking diversity in this context means a dataset that has considerable data for one group, but does not have many examples for another group. Nikon and Hewlett Packard's cameras’ being trained mainly on white people resulting in them not being able to recognize people of color effectively is an example of this type of bias. Governor neural networks preprocesses data before feeding it into a neural network. The purpose of this preprocessing is identifying archetypes in the data. Archetypes refers to paradigms found between the variables of each input. Once these archetypes are identified governor neural networks will alter the dataset so that the network is seeing an equal number of examples for each archetype (Blank, 2004). For example, if four archetypes are identified then the dataset will consist of 25% data for each type, and would then proceed to train the neural network on this data. This algorithm both identifies, and is a possible solution to the issue of selective bias. However, since it is generating the same data from a single example, an issue with this approach is that it would not show diverse examples within a subgroup. For example, suppose Nikon trained its cameras on mostly white people and a few mixed raced people. Using governor neural networks to produce a training dataset that consists of 50% white people, and 50% mixed raced people would not be effective at identifying people of all races. The 50% consisting of mixed race cases would be too similar, since it is replicated data from only a few examples. However, it will be more effective than had governor neural networks not been used at all.

2.5 Case Study - Recidivism

Artificial Intelligence systems could take on a larger role in parole and sentencing decisions by predicting a criminal's chances of being a repeat offender (Angwin, 2016, Article 1). The results of neural networks that predict a criminal's chances of recidivism are already being considered by judges in Arizona, Colorado, Delaware, Kentucky, Louisiana, Oklahoma, Virginia, Washington, and Wisconsin, and could potentially be expanded to more states (Angwin, 2016, Article 1). The creators of these networks claim that it is not racially biased, because it does not use zip codes or race as variables of consideration in its algorithm (Coane, 2016). Given the history of racially segregated neighbors in America through redlining, the decision to exclude zip codes as a variable makes sense. However, income and associations are included variables in the algorithm, which are factors related to race given income inequality, and cultural segregation in America. The inclusion of these variables has the potential to create racial biased and classist conclusions. Statistically, people of lower incomes are more likely to be imprisoned for crimes committed, in contrast to people of higher incomes who have the advantage of paying for more experienced lawyers to represent themselves. Thus, including jail time as a factor of recidivism without considering income as well, could reflect wealth and not recidivism risk, resulting in incorrect predictions.

ProPublica did a critique of Northpointe's "Correctional Offender Management Profiling for Alternative Sanctions" software or COMPAS software using recidivism data from Broward County, Florida (Angwin, 2016, Article 1). It was found that Northpointe predicted recidivism correctly only 61% of the time, meaning more than one in three cases are judged incorrectly by the software. The results from COMPAS demonstrated significant racial bias, and mislabeled black people as "Higher Risk" who did not end up re-offend 44.9% of the time, and in the case of white people this issue happened only 23.5% of the time. These results mean that black people were mislabeled as high risk twice the amount that white people were (Angwin, 2016, Article 1). These bias statistics probably do not arise from selective bias considering ProPublica used 10,000 cases in its review of COMPAS, but in variable bias, using variables that are strongly affected by racism.

The proponents of using neural networks to predict recidivism claim that this program provides "just information" to help prosecutors with their decision (Coane, 2016). However, it is impossible to tell how dependent prosecutors could potentially become on this software to make decisions. Some critics also argue that the use of software based on statistics in courtrooms is unconstitutional, because it defeats the purpose of considering the individual in a legal setting.

This case study exemplifies the complex issues surrounding the integration of artificial intelligence into our everyday lives, especially to what extent data should be used to decide the opportunities presented and provided to us.

3.1. Methodology – Proxy Variables

Proxy variables are variables for which we would theoretically not want the network to take into consideration in its decisions. Depending on the purpose of the network, race or gender could be undesired variables that we would not want the network to take into consideration. Proxy variables are variables that act as a proxy for an undesired variable in a network.

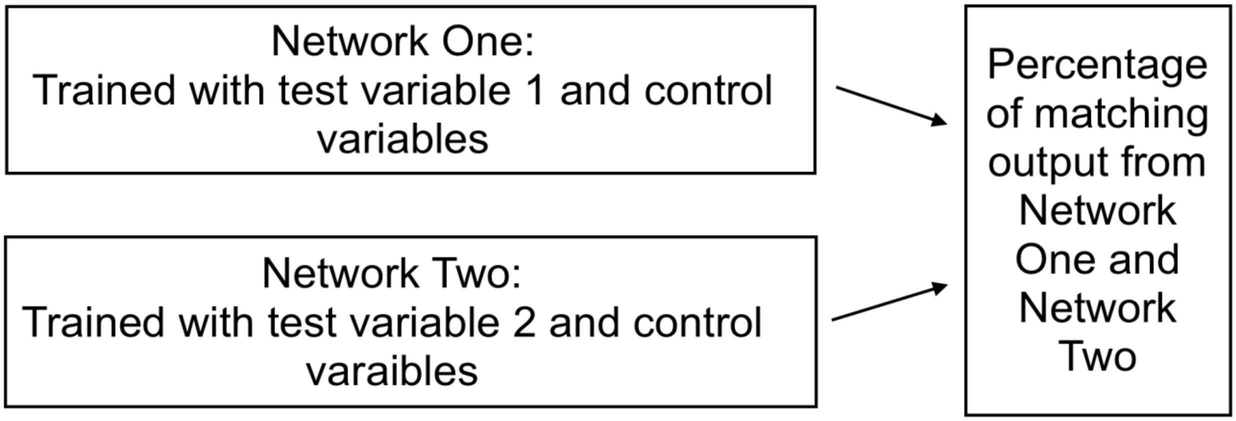

Network One:
Trained with test variable 1 and control variables
Network Two:
Trained with test variable 2 and control varaibles
Percentage of matching output from Network One and Network Two

Figure 1: Proxy Variable Flow Chart

To identify proxy variables in a training dataset, two networks can be trained. In Figure 1 these two networks are represented as Network One and Network Two, where Network One contains one test variable and Network Two contains another test variable. The networks are compared by propagating using the same test data, apart from using either test variable 1 or test variable 2. The last step is to compare the correlation of the test results from these two networks. If they both have the same test result, the count is incremented. Finally, this count is divided by the size of the test dataset to give a proxy percentage or measure of similarity between the results of the two networks. This percentage will always be bellow one hundred percent, because it is impossible for the two networks to have more agreements in results than there is data to test on. The higher the percentage, means that there is an increased likelihood that test variable 1 and test variable 2 are acting as proxy variables, and conversely the lower the percentage indicates less correlation between the two variables. If the proxy percentage is close to a hundred percent, and one of the two variables that a network was trained on is an undesired variable to include in the decision-making process, it can be concluded that the second variable should not be included either.

3.2. Methodology – Measuring Equality

Measuring equality is a technique that examines a subgroup of data and compares its results to them as a whole group, in order to get an indication of selective bias towards a subgroup. Selective bias is a pro or con bias towards a subgroup or minority, called the bias ratio. The bias in the network being tested would not stem from the machine learning algorithm, but from the data inputted in the training dataset causing it to make skewed conclusions.

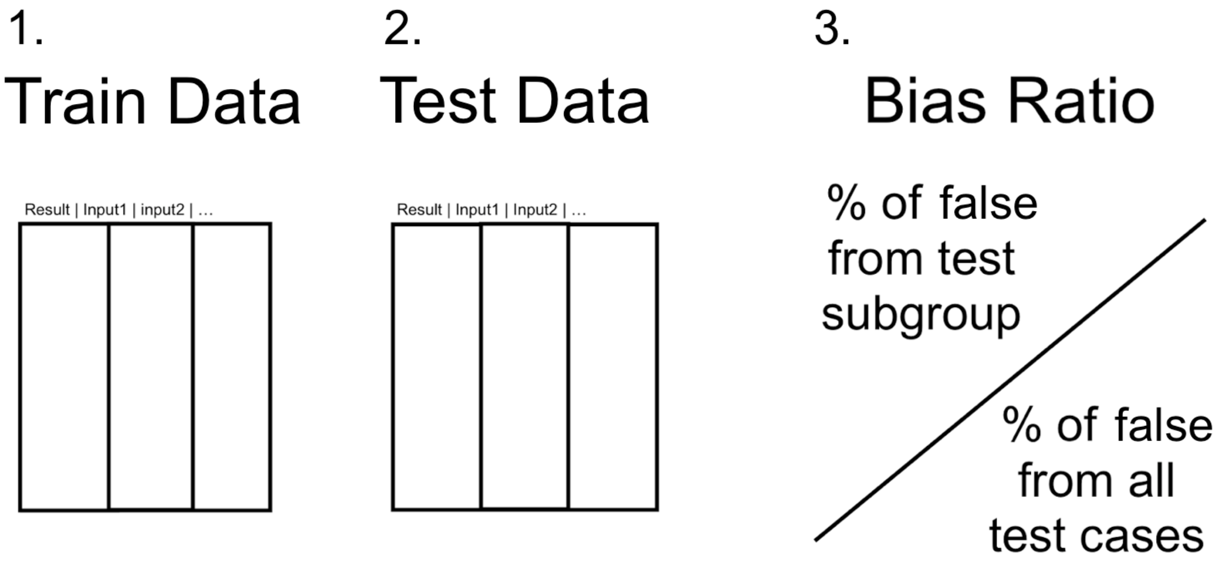

1.
Train Data
Result | Input1 | input2 | …
2.
Test Data
Result | Input1 | Input2 | …
3.
Bias Ratio
% of false from test subgroup
% of false from all test cases

Figure 2: Steps to obtain Bias Ratio chart

The bias ratio in Figure 2 compares the variance of the subset to the overall group, and gives an indication of the positive or negative bias the network has for the subset. Training data, test data, and resulting data from the network are all used to compute a bias ratio. Comparing the conclusions, a network makes overall, in contrast to subsets of the data provide insight into the frequency of discriminatory conclusions made by a network.

4.1. Results and Analysis

Both techniques to quantify bias that used the data described above were coded in Python using a Neural Network library named Conx which is built on top of the Theano library (Theano, 2016). Theano is Python library that specializes in the defining, optimizing and evaluation of multidimensional arrays. Multidimensional arrays are a vital component for processing the input, hidden layer, and output of neural networks (Theano, 2016).

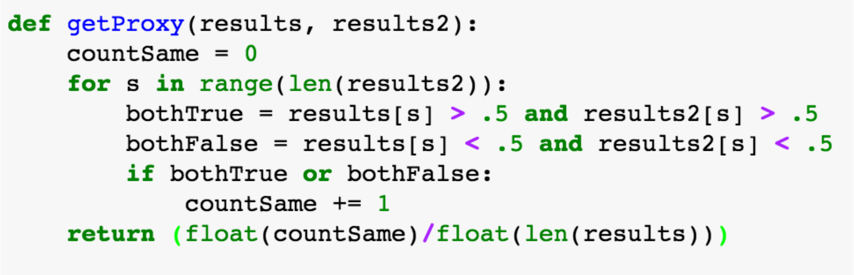
Both of networks used to implement proxy variable and measuring equality take in multiple inputs ranging from 9 to 13, have a hidden layer set to size 60, and output a single number between 1 and 0. These parameters are defined in the declaration of a network “net = Network(8, 60, 1)” where “net” is the neural network variable. The network produced is a supervised network, meaning that it is taught desired output from a training dataset.

The dataset used to train all networks is a COMPAS scores file named “compass-scores-two-years.csv” on GitHub (Larson, 2016). Part of this dataset was used to inform COMPAS scores, and includes the generated scores, but does not contain all variables that generated the COMPAS scores. This dataset was used in ProPublica’s article criticizing Northpointe and Northpointe’s response publication to ProPublica’s article. This file is table of results from Northpointe’s software including if the criminal to recidivate within two years of being released. It does not include all Northpointe’s inputs, because they are not public. The results in this paper are a tertiary analysis of Northpointe’s software results, and not a recreation of Northpointe’s software.

There were six variables from the COMPAS scores file used for the creation of the networks for this project including, sex, age, race, risk of violence score, risk of recidivism score, and if recidivism happened with two years of release. Sex was mapped to a single input, 1 for male, and 0 for female. Age could have been mapped to single input between 1 and 0, however to help the network recognize patterns age was instead mapped to three inputs either 1 or 0 for each indicating “less than 25,” “25 - 45,” and “greater than 25.” Only races listed as “White” (0) and “Black” (1) were considered in the network. Risk of violence score, and risk of recidivism score were represented in the same way with three inputs corresponding to “Low,” “Medium,” and “High” risk. There is one exception to this, in the network used to predict risk of recidivism scores the target value was set to either 0, .5, or 1 indicating “Low,” “Medium,” and “High” risk. Both the risk of violence and risk of recidivism result from Northpointe’s COMPAS network. Lastly, recidivism within two years of release was represented as 1 for recidivism and 0 for no recidivism.

4.2. Identifying Proxy Variables - Results and Analysis

The implementation of identify proxy variables assumed that there was a single output from a network, where outputs bellow .5 indicates false, and above .5 indicates true. The code for identifying proxy variables can be seen bellow in Figure 3. The “getProxy” function in this code takes in the results from Network 1 and Network 2 correspondingly saved in arrays named “results” and “results2.” It considers each element of the results, and counts how many are the same. Finally, it returns a proxy percentage indicating the level of correlation between the two variables.



```
def getProxy(results, results2):
    countSame = 0
    for s in range(len(results2)):
        bothTrue = results[s] > .5 and results2[s] > .5
        bothFalse = results[s] < .5 and results2[s] < .5
        if bothTrue or bothFalse:
            countSame += 1
    return (float(countSame)/float(len(results)))
```

Figure 3: getProxy function

Three different tests to identify proxy variables were run. For all three of these tests the target output was if the criminal recidivated within two years of release. The percentage correctness of these networks ranged from .49 to .51, except for the network excluding the risk of violence variable which lowered its percentage correctness to .41. All six networks were trained with 4000 rows of data that only included cases of Caucasian and African American, 1500 epochs of training, and a tolerance of .4. Figure 4 displays the three proxy percentages produced ranging from lowest to highest percentage.

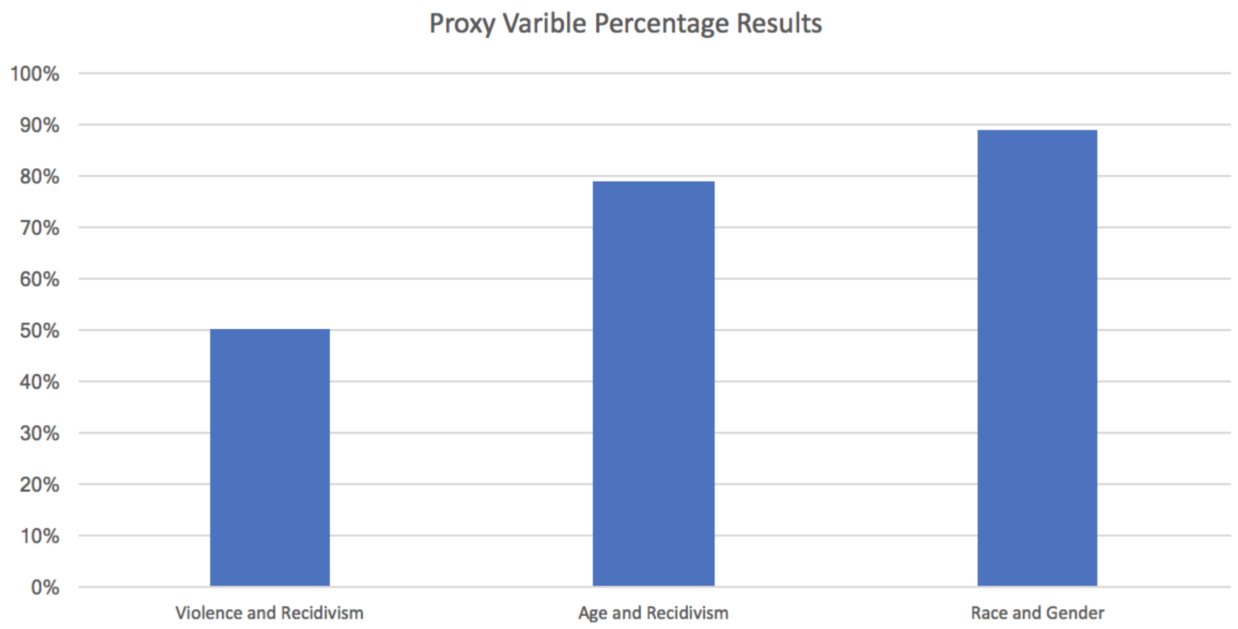


Figure 4: Proxy Variable Percentage Results from 6 networks

My hypothesis for the first second test where variable 1 and variable 2 were set to risk of violence and risk of recidivism, because they are relatively similar variables, and thus proxy variables having a proxy percentage close to 1. This hypothesis failed, since the resulting proxy percentage was just below 50%, the lowest out of all three tests when I expected it to be the highest. The lower correctness percentage of 41% in the network excluding the risk of violence variable most likely had an influence on this outcome. The conclusion from this test is that risk of violence heavily weights the decisions of the network.

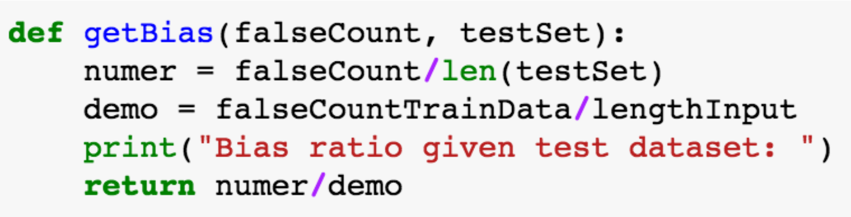
The hypothesis for the second test where variable 1 and variable 2 were age and risk of recidivism, because they are unrelated variables. I suspected that age has little effect on the outcomes of the network, and recidivism would intuitively have more influence. Furthermore, I supposed that age and risk of recidivism are uncorrelated variables, and would not produce similar results. This hypothesis was not confirmed as can be seen by the bar percentage named Age and Recidivism, having proxy percentage larger than Violence and Recidivism.

Lastly, the hypothesis for the third test where variable 1 and variable 2 were race, and gender was parallel to test two. Just as in test two, I supposed that they are relatively unrelated variables, and thus have a proxy percentage closer to 0. However, just as in test two this was found to be incorrect, the resulting proxy percentage was 88% the highest out of all three tests.

A disadvantage of the proxy variable technique is that it generally ignores the fact that trained networks do not equal weight each variable. I believe this is the main reason all the hypothesis failed, since risk of violence seems to play a vital role in the accuracy of the network. Thus, when testing two variables that are highly correlated to each other, but are also highly correlated to the outcome, their proxy percentage could be lower than expected. In both cases the networks bases its conclusions heavily on the value of the variable and any difference between the two variables will be exaggerated in the resulting proxy percentage whenever there is variance in input. However, this technique was successful in identifying risk of violence as a more influential variable than risk of recidivism. This indicates that the COMPAS network is highly inaccurate when predicting risk of recidivism, but that their risk of violence score does have an indication if a criminal will recidivate within two years.

4.3. Measuring Equality - Results and Analysis

The implementation of measuring equality measures the parity of output from a neural network. This implementation measuring equality assumes that there is a single output from a network, where outputs bellow .25 indicates “Low,” .25-.74” indicates “Medium” and above .75 indicates “High”. The target values from the training dataset are either 0 indicating “Low,” .5 indicating “Medium” or 1 indicating “High.” The code for measuring equality can be seen bellow in Figure 5. The “getBias” function in this code takes in “falseCount,” the count of false results in a test dataset and “testSet,” which is the test dataset that is used for its length.



```
def getBias(falseCount, testSet):
    numer = falseCount/len(testSet)
    demo = falseCountTrainData/lengthInput
    print("Bias ratio given test dataset: ")
    return numer/demo
```

Figure 5: getBias function returns a bias ratio

The variable “numer” in Figure 5 is the percentage of false results from a network and the variable “demo” is the fraction of false instances from the training dataset. Lastly, “numer” is divided by “demo,” which gives the percentage of false from the test dataset in compassion to the percentage false in the dataset the network was trained on. If the test dataset consists of a subgroup category that we wish to compare to how the network handles in comparison to how it handles data from all groups, this fraction will give a measurement of that bias called a bias ratio. This bias ratio can be used to identify selective bias by giving a measurement of equality.

The measuring equality technique was implemented on a single network. The network was trained with 4000 rows of data that only included cases of Caucasian and African-American, 1500 epochs of training, and a tolerance of .25. The percentage correctness of these network was 65%, indicating that predicting the recidivism score is easier to predict than actual recidivism giving the risk of recidivism score as an input variable. Predicting COMPAS output is a simpler problem than predicting if a criminal will recidivate. Thus, it can be concluded that the COMPAS network is an insufficient model for the problem it attempts to solve. Figure 4 displays the three proxy percentages produced ranging from lowest to highest percentage.

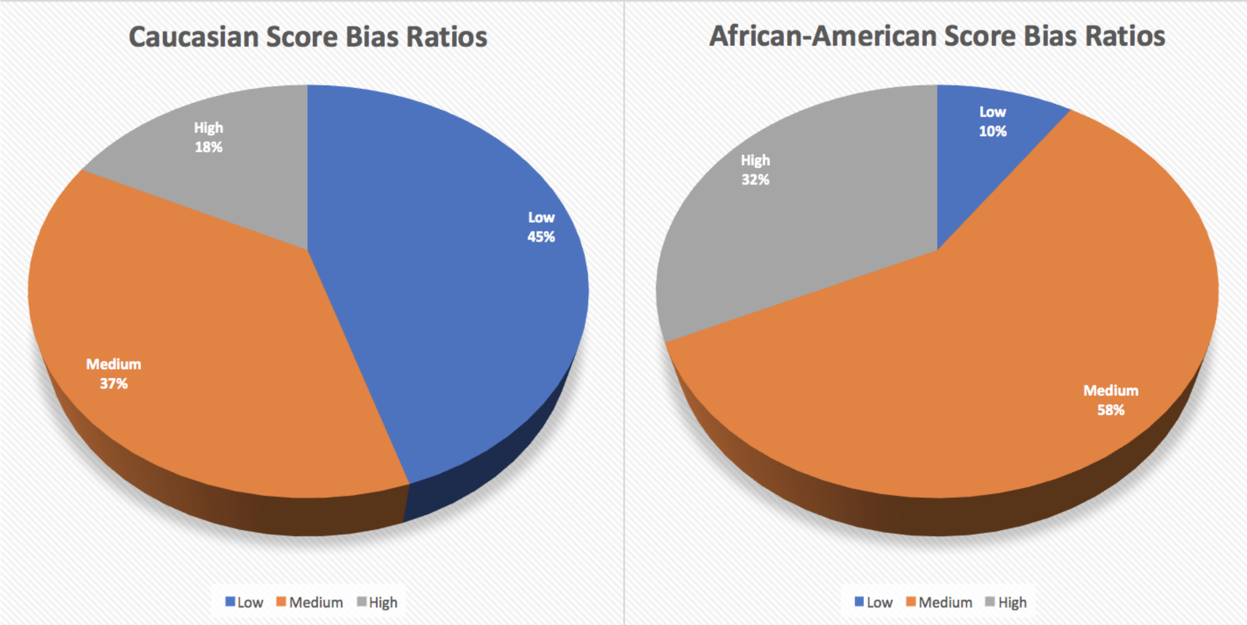


Figure 6: Selective Bias results from network with recidivism scores as the target and two altered test datasets one Caucasian and the other African-American

The results in Figure 6 were generated from two test datasets that used the remaining 2149 rows of COMPAS data that the network was not trained on. The two datasets are identical except that they were both altered so that value was all Caucasian in one and all African American in the other. The motivation for doing this was to test how the network is influenced by race.

In general, the medium range contains the most data which makes sense since its range is from .25 to .75 making it twice as large as the Low and High classification. In hinds sight this is a flaw in implementation, I should have evenly divided ten into three parts adding 8% of results to Low and High from Medium. This mistake would not have greatly effected the trends seen in the high and low outputs since they share similar bias ratios. An extra 8% of data moved from the Medium score to Low score could have helped to close the preference for Caucasians in the Low category, but it would not have been enough to significantly change the trend shown above. Another caveat to these results is that they were not trained on all the inputs Northpointe used, such data is not available publically. It is possible that this data could have produced a less biased network had it been used. Although I suspect this to be unlikely because some of the other questions considered by Northpointe are “How often did you get in fights while at school?” and “was one of you parents ever sent to jail or prison?” (Angwin, 2016, Article 1). These questions are potentially another way of asking a person’s race, rather than a way attaining substantial knowledge of their chance of recidivating.

The results shown in Figure 6 indicate that the Low label is assigned to Caucasian people more than four times more often than African-American. The Medium label is assigned to African-Americans much more than Caucasians. Lastly, the High label is assigned to both races less frequently, but it is assigned to African-Americans almost twice as frequently as Caucasian people. Thus, the ideal Low scores are assigned to Caucasian and the unideal Medium and High scores are assigned to African-Americans much more often when the only difference between criminals is their race. These results show that there is blatant bias against African-Americans in the trained neural network.

5. Future Work

The issue of cognitive bias in neural networks is an interdisciplinary one involving large datasets and various sociological factors. Identifying proxy variables would ideally be used by data scientist in conjunction with sociologists, lawyers, and psychologists to make conclusions on which variables should be included in the process of training neural networks. As seen by the implementation of identifying proxy variables and measuring equality, it is hard to interpret the results without social context. An interdisciplinary approach is potentially very slow and expensive since it would most likely require the gathering of more data to offset bias once an approach has been decided upon. Furthermore, since identifying proxy variables is done after a network is run each addition of new data would be added to the training set of the network. It cannot be predicted exactly how this new data will affect the quality of the outcomes from a network.

The tools for measuring equality in this project could be used any learning system. Measurements of bias can be used to inform the reweighting of fields used in a neural network so that the network bases its conclusions more or less heavily on different variables. A backpropagation approach would be ideal for networks where the goal is to have even distribution for all relevant subsets of the data.

6. Conclusions

It is impossible to quantifiably conclude that human bias doesn't exist or can truly be unlearned from an individual, just as is impossible to make these claims for an artificial intelligence network. Nor would we want to claim this because the value of artificial intelligence is that it has opinions or in other words bias to make informed decisions. However, this does not mean that this issue cannot be quantified or should be left unexamined. If this problem is left to fester, discrimination will continue living on in digital platforms.

References

1. (Angwin, 2016, Article 1)
Angwin, Julia, Jeff Larson, Surya Mattu and Lauren Kirchner. “Machine Bias” *ProPublica* 23 May 2016, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing Accessed 29 Jan 2017.

2. (Angwin, 2016, Article 2)
Angwin, Julia, Jeff Larson, Surya Mattu and Lauren Kirchner. “How We Analyzed the COMPAS Recidivism Algorithm” *ProPublica* 23 May 2016, https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm Accessed 29 Jan 2017.

3. (Blank, 2004)
Blank, Douglas, Jeremy Stober and Lisa Meeden. “The Governor Architecture: Avoiding Catastrophic Forgetting in Robot Learning” *Bryn Mawr and Swarthmore College*, 2004, https://www.researchgate.net/publication/241568133_The_Governor_Architecture_Avoiding_Catastrophic_Forgetting_in_Robot_Learning Accessed 20 Mar 2017.

4. (Caliskan-Islam, 2016)
Caliskan-Islam, Bryson, and Narayanan. “Semantics derived automatically from language corpora necessarily contain human biases” *Princeton Uni*versity, 30 Aug 2016, http://www.princeton.edu/%7Eaylinc/papers/caliskan-islam_semantics.pdf Accessed 8 Dec 2016.

5. (Cherry, 2017)
Cherry, Kendra. “What Is a Cognitive Bias? Definition and Examples” *Verywell*, 9 May 2016, https://www.verywell.com/what-is-a-cognitive-bias-2794963 Accessed 20 Jan 2017.

6. (Coane, 2017)
Coane, Marty Moss. “Using crime predictors in criminal justice” *WHHY: RadioTimes with Marty Moss Coane*, 14 Nov 2016, http://whyy.org/cms/radiotimes/2016/11/14/using-crime-predictors-in-criminal-justice Accessed 20 Jan 2017.

7. (Collins, 2016)
Collins, Nathan. “Artificial Intelligence Will Be as Biased and Prejudiced as Its Human Creators” *Pacific Standard*, 1 Sep 2016, https://psmag.com/artificial-intelligence-will-be-as-biased-and-prejudiced-as-its-human-creators-38fe415f86dd#.gouhk79ow Accessed 1 Nov 2016.

8. (Crawford, 2016)
Crawford, Kat. “Artificial Intelligence’s White Guy Problem” *New York Times*, 25 June 2016, http://www.nytimes.com/2016/06/26/opinion/sunday/artificial-intelligences-white-guy-problem.html?_r=0 Accessed 1 Nov 2016.

9. (Dieterich, 2016)
Dieterich, William, Christina Mendoza, and Tim Brennan. "COMPAS Risk Scales: Demonstrating Accuracy Equity and Predictive Parity" *Northpointe Inc. Research Department*, 8 July 2016, http://go.volarisgroup.com/rs/430-MBX-989/images/ProPublica_Commentary_Final_070616.pdf Accessed 10 Mar 2017.

10. (Ingold, 2016)
Ingold, David and Spencer Soper. "Amazon Doesn't Consider the Race of Its Customers. Should It?" *Bloomsburgh*, 21 April 2016, https://www.bloomberg.com/graphics/2016-amazon-same-day/ Accessed 20 Mar 2017.

11. (Larson, 2016)
Larson, Jeff. "Propublica/compass-analysis" 29 Jul 2016, https://github.com/propublica/compas-analysis Accessed 1 Mar 2017.

12. (Leverington, 2015)
Leverington, David. "A Basic Introduction to Feedforward Backpropagation Neural Networks" *Texas Tech University*, 2009 http://www.webpages.ttu.edu/dleverin/neural_network/neural_networks.html Accessed 1 Apr, 2017.

13. (Pennington, 2015)
Pennington, Socher, and Manning. "GloVe: Global Vectors for Word Representation" *Stanford University*, October 2015, http://nlp.stanford.edu/projects/glove/ Accessed 1 Nov 2016.

14. (Rose, 2010)
Rose, Adam. "Are Face-Detection Cameras Racist?" *Time*, 22 Jan 2010, http://content.time.com/time/business/article/0,8599,1954643,00.html Accessed 20 Jan 2017.

15. (Rumelhart, 1986)
Rumelhart, David and James McCelland. "The Appeal of Parallel distributed processing" *MIT Press*, 1986 https://stanford.edu/~jlmcc/papers/PDP/Chapter1.pdf Accessed 20 Jan 2017.

16. (The Theano Development Team, 2016)
The Theano Development Team. "Theano: A Python framework for fast computation of mathematical expressions" May 9 2016, https://arxiv.org/pdf/1605.02688.pdf Accessed 10 Mar 2017.