Learning Scoring Systems with Deep Rectifier Neural Networks

Greg Carmean gcarmea1@jh.edu
Engineering For Professionals
Johns Hopkins University
Baltimore, MD 21218-2608, USA

Abstract

This paper explores how to learn a scoring system that represents with full fidelity a deep rectifier (ReLu) neural network. Scoring systems are a small set of weighted features that when added or subtracted and compared to a threshold can be used to make a transparent classification decision. These are used in high stakes applications where interpretability concerns currently limit the use of deep neural networks. The scoring system can be generated by approaching the construction of the system as a minimization of its weights with the trained neural network used as a constraint set(Márquez-Neila et al., 2017), and exploring the relevance of the inputs through relevance aggregation,(Grisci et al., 2021). Experiments show that this approach can generate interpretable scoring systems that in some cases match the accuracy of the original neural network.

1. Introduction

Neural networks have become an integral tool in machine learning in low stakes applications such as computer vision, speech recognition and natural language processing. Their ability to obtain high classification accuracy has contributed to their adoption in those fields. However, an impediment to their adoption in other fields that involve more high stakes decision making is the fact that the models are black-box. The user has little to know understanding on how the prediction is generated, which limits the utility of these models in certain areas. The prevalence of highly accurate but opaque models has lead to the idea of a tradeoff between accuracy and explainability in machine learning.

The lack of interpretability for these black box model has led to their exclusion from applications that require interpretability, such as financial medical and criminal justice. For example, a reason must be given to explain the decision whether or not to grant a loan. Criminal justice decisions must stand up to legal challenges, and doctors need to understand why a model is flagging their patient for a treatment in order to give them the best care. All of these applications require interpretable models that subject matter experts can interact with and understand how the model reaches a classification decision.

Scoring systems are an example of a linear model currently successfully deployed in these environments. Scoring systems consist of features that can be scored with small digits and where the addition, subtraction or multiplication of those numbers leads to a score which maps to a classification decision. Finance applications that use these include the FICO credit score (FICO, 2011) and models that inform investment decisions (Beneish et al., 2013). In the criminal justice world these models, such as ORAS (Latessa et al., 2009) are used to predict recidivism among detainees. Finally, there are several scoring systems widely used in medicine today, such as the DiaRem score(Still et al., 2014) the $CHADS_2$ index (Gage et al., 2001) and SAPS3 (Moreno et al., 2005). DiaRem is used to predict pre-operation the remission of diabetes of patients getting ready to undergo gastric bypass surgery. The $CHADS_2$ system helps physician assesses the risk of stroke in patients with attrial filbrulation, and SAPS3 helps assess the mortality risk for patients in intensive care. These models have been successful not only because of their accuracy but because of their transparency and ease of use by practitioners. They are also, in some cases such as $CHADS_2$ and SAPS, developed by hand and not optimized, presenting an opportunity for machine learning to help develop better models.

This paper builds on previous work in using machine learning to develop these scoring systems, presenting a way for these models to be constructed from deep rectifier (ReLu) neural networks. This could help bridge the gap between neural networks and high stakes applications. Recent improvements in rectifier neural network interpretability can be leveraged to enforce sparseness in the model, and the feature weights can be optimized with the trained network as constraints. The hypothesis is that a neural network can be generated to produce a interpretable decision scoring system with comparable accuracy to normal neural network performance.

2. Related Work

Scoring Systems: The idea of using machine learning to develop scoring systems is not a new one. The Supersparse Linear Integer Model (SLIM) (Ustun et al., 2013) is an off the shelf tool proposed in 2013 for developing scoring systems for decision classification tasks. The paper outlines quantifiable traits that help a scoring system be intepretable, such as being constrained to find super-sparse results with small digits coefficients, and the ability to add constraints from domain knowledge. The product of the research, SLIM, is an off the shelf tool for constructing optimal scoring system that have those characteristics. Further work was done by these authors in proposing Risk-Calibrated Supersparse Linear Integer Model (RiskSLIM)(Ustun and Rudin, 2019). Risk scores are a more dynamic then decision criteria as do not provide a yes or no response but measure risk on a scale. The authors proposed a new cutting plane algorithm for the efficient calculation of the MINLP problem that risk score systems can be represented as, to aid in the efficient computation of those results.

Other work has been done in learning scoring systems. The Fully Corrective Binning algorithm (Sokolovska et al., 2018) is a method to simultaneously bin continuous attributes and learn the weights associated with those bins. The authors prove the approach can generate an optimal solution for a scoring system with a given set of constraints, and it has the advantage of providing the optimal binning of the features, which can be useful for learning diagnostic or other important thresholds from the data. Another approach to learning a scoring system is given in the construction of DILSVM. (Carrizosa et al., 2016). The feature values for the problem are constrained to a preset number, so that the output matches a Likert scale representation of whether the feature agrees or disagrees with the class. The resulting model is shown to have comparable accuracy with the regular SVM models of similar datasets.

Interpretability in Deep Neural Networks: Ever since neural networks and deep learning began to grow in popularity, there has been an explosion of interest in better interpreting and explaining the behavior of those models.(Molnar et al., 2020). There are several methods such as Shapely Values, which help explain how features contribute to a prediction, and counterfactual examples (Wachter et al., 2017), which help explain individual predictions. Another methods for both a local and more global interpretation of neural networks has been to build prototypes of the concept learned by the network. By communicating how similar the output is to the prototype, the model can show what part of an input was involved in its classification decision.(Montavon et al., 2018),(Arık and Pfister, 2020).

Another approach to interpretability has been to take advantage of the graph structure of a deep neural network directly via a method called Layer-wise Relevance Propagation (LRP)(Montavon et al., 2018). LRP first propagates out from the input layer to the output layer, and then works backwards from the output neuron to the input layer to trace the relevance back through the network, following a local propagation rule to conserve the relevance throughout the layers, similar to local Taylor Decomposition. Relevance is conserved throughout each layer of the network. The relevance propagation rule (Montavon et al., 2019) has the form

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k$$

Where j and k are neurons at two consecutive layers, R_k is the relevance score of the k neuron, and $z_j k$ captures the contribution j makes to k's relevance. In the ReLu networks used in this paper,

$$z_j k = a_j w_{jk}$$

Where $a_k = max(0, \sum_{0,j} a_j w_{jk})$ or the lower layer activation of the neurons plus the bias neuron w. For the initial layer $a_0 = 1$. The possible relevance propagation rules all have the form

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$

The precise propagation rule selected is an experimental design choice described in section 4.

The LRP algorithm has recently been extended in (Grisci et al., 2021) to construct a relevance aggregation algorithm for ReLu neural network interpretability involving tabular data. The aggregations involved in relevance aggregation is able to overcome some of the challenges introduced by the fully connected layers present in networks constructed for tabular data that LRP struggles with on a per sample basis. The aggregation algorithm determines which features of the network are most relevant towards the output over the entire set of training data. For multi-class data, the relevance is provided globally as well as by class. The single relevancy score from 0 to 1 not only gives insight into how the trained neural network behaves but can be used as a feature selection tool as well.

3. Scoring System Learning Approach

Background: Previous work in developing optimal scoring systems (Ustun et al., 2013), (Sokolovska et al., 2018) has shown how the development of a scoring system can be framed as an optimization problem minimizing the loss function subject to the constraints that make the problem interpretable. Using the framework presented(Ustun et al., 2013), the optimization can be set up as follows. Considering linear classifiers in the form of $\hat{y} = sign(x^T\lambda)$, where $x \in X \in \mathbb{R}^P$ is a vector of the p features of the dataset, $\hat{y} \in Y = -1, 1$ represent the predicted label and $\lambda \in \mathbb{R}^P$ represent the coefficients to be learned. The training data with N examples $(x_i, y_i)_{i=1}^N$, and the set of interpretability constraints $\mathbb{L}$ can be combined to form the following optimization problem:

$$min_\lambda \; Loss(\lambda; (x_i, y_i)_{i=1}^N) + C \cdot InterpretabilityPenalty(\lambda)$$

$$s.t \quad \lambda \in \mathbb{L}$$

This type of Mixed Integer Linear Programming problem (MILP) can be solved by neural networks by using a ReLu network as a surrogate for the MILP and optimizing the weights of the neural network(Márquez-Neila et al., 2017). The ReLu network has K+1 layers from 0 to K, where 0 is the input layer, K the output layer and the middle layers all correspond to hidden layers. Marquez (2017) shows that with the introduction of a binary activation variable z, the ReLu logic can be explicitly programmed via big-M constraints. The output for a given layer can be decomposed into the equation

$$w^T g + b = h - s$$

where w represents the layer weights, h represents the positive part of the output and s represents the negative part of the output. With finite values of L and U such that $L \leq w^T g + b \leq U$ then the big M constraints are

$$h \leq Uz$$

$$s \leq -L(1 - z)$$

$$z \in (0, 1)$$

Under this formulation for each node (j,k) the ReLu network can be written as

$$Input\ Layer$$

$$L^0 \leq h^0 \leq U$$

$$Hidden\ (ReLu)\ Layers$$

$$W^k h^{k-1} + b^k \; for \; k = 1, ..., K - 1$$

$$h^k \ , \ s^k \geq 0 \; for \; k = 1, ..., K - 1$$

$$z^k \in (0, 1)^{n_k)} \; for \; k = 1, ..., K - 1$$

$$x^k \leq U^k z^k \; for \; k = 1, ..., K - 1$$

$$s^k \leq -L^k(1 - z^k) \ for \ k = 1, ..., K - 1$$

$$OutputLayer$$

$$W^K h^{K-1} + b^K = h^K$$

$$L^K \leq h^K \leq U^K$$

Relaxing the z definition to $0 \leq z \leq 1$ is the ReLu relaxation which allows a variety of bound tightening algorithms to be applied. The Iter-RR algorithm defined by Marquiz (2017) will be used in this approach. The algorithm works by iterating through each layer and each node in the layer. The bounds from the previous layer are propagated to the next and used to solve for the upper and lower bound respectively for each node. The mean absolute distance between the bounds is calculated and the compared to a threshold. The process continues until the distance between the bounds is below the threshold or a set number of iterations has elapsed.

Approach: Employing the formulation from (Márquez-Neila et al., 2017) allows a trained network to be constructed as constraints in another optimization problem. Therefore, in this approach the objective of the minimization is to impose sparseness in the feature weights with the l_1 norm.The loss function, cross-entropy, is already minimized in the training of the network and the network constraints ensure the loss stays minimal when creating the scoring system. Additional interpretability constraints will include a tuneable minimum relevancy aggregation score for the features that remain in the network. The model can then be written as

$$min_\lambda \ \frac{1}{N} \sum_{n=1}^{N} \ \parallel \lambda \parallel_0$$

$$s.t \ trainedReluNetworkConstraints$$

$$Relevancy(\lambda) > C_1$$

$$|\lambda_i| < 100 \ for \ i = 1, ..., P$$

$$|\lambda_i| >= 0 \ for \ i = 1, ..., P$$

$$\sum_{n=1}^{N} \lambda >= C_0 for \ i = 1, ..., P$$

C_0 represents a tuneable parameter to control the sparseness of the network and C_1 controls the relevancy of the features towards a prediction. The aggregate relevance represents an additional way to ensure the network is sparse and that all of the final features contribute meaningfully to the scoring system. The constraint that the coefficient values be under 100 is added to ensure that the results of the scoring system are small enough to be easily interpretable, and constraining the weights to be positive makes the minimization practical. Although not explored in these experiments, an advantage to this approach is that problem specific constraints driven by situational requirements or prior knowledge can be added to the optimization problem. This could help improve the performance of the resulting scoring system.

The weights produced from solving the optimization problem becomes the weights of the features in the scoring system. For continuous features, the weight is multiplied by the feature value to determine the score for that feature. For categorical features, the weight is the value added if the feature exists, and 0 otherwise. The threshold for classification decisions is tuned from the training data. This implementation of the approach uses the mean of the resulting scores as the yes/no classification threshold for consistency across experiments in instances. In the instance of a single binary feature, a positive is considered a positive classification. However, how the threshold is determined could be further adjusted for particular problem domains where avoiding false negatives or false positives is more important, to reduce those occurrences in the final scoring system.

4. Experiments

Experimental Design: The hypothesis is tested by evaluating the accuracy, fidelity, interpretability and consistency of the scoring systems built via the approach. For each dataset considered, the accuracy is assessed via average classification accuracy of the scoring systems produced from 10-fold cross validation. The accuracy of the full fitted neural network is also assessed on the same testing data to determine the fidelity of scoring system to the original model. The interpretability of the scoring systems are assessed by how closely they followed the desirable traits of a interpretable systems: sparsity and their intuitiveness. Scoring systems will be deemed sparse if they have a maximum of 7 feature (Ustun et al., 2013) in the resulting scoring system. The intuitiveness of the scoring system will be measured by the whether the resulting coefficients can be represented in 1-3 significant digits. Finally, the scoring systems produced for the same dataset should be consistent. If the algorithm is learning a general scoring system, then the scoring system generated across cross validation folds should be similiar if not identical.

Scoring systems were generated for four different datasets, 3 from from the UCI Machine learning repository (Dua and Graff, 2017), the breast cancer(Mangasarian and Wolberg, September 1990) , adult income and bank marketing (Moro et al., 2011) datasets, and the glaucoma dataset from the "ipred" R package. The breast cancer dataset contains 699 observations of either benign or malignant tumors, with 10 real valued attributes associated with each observation and a roughly even class distribution. The adult income data contains 48,842 records of adults in the United States 1994 census, with the prediction objective of determining whether the adult has an income over or under 50k, with only 24% of the dataset over 50k. The glaucoma dataset has 170 observations with 66 attributes and an evenly distribution between normal and glaucoma samples. Finally, after one hot encoding the categorical variables, the bank marketing data has 48 attributes and 88% of the samples are no. Together these four datasets cover a range of number of attribute and class imbalances, as well as combining categorical and discrete features.

Each dataset is modeled by a neural network with three hidden layers, whose layer sizes are tuned before constructing the scoring system. Keras was used to construct and train the ReLu neural networks. The cancer dataset neural network includes hidden layers with 8 and 5 nodes, the bank dataset neural network includes hidden layers with 21, 11 and 5 nodes, the adult dataset neural network includes hidden layers with 29, 14 and 7 hidden layers, and finally the glaucoma dataset neural networks have hidden layers with 26,16 and 8 hidden layers. Softmax is used as the output layer activation function. Relevance aggregation is performed on the tuning neural network to determine the threshold to apply for inclusion into the scoring system. One design decision made in the use of relevance aggregation was the choice to use the LRP-ϵ rule in the relevancy aggregation calculation. The ϵ introduced to the rule can offset contradictory relevance signals into the layer. Of all of the rules described by Montavon et al., LRP-ϵ promotes the most sparsity in the relevant features returned, which is desirable for this application. The ϵ value used in the experiments is 0.01. Tuning showed that the relevance scores were fairly insensitive to changes in ϵ selected for the examined datasets.

After the network is trained, the optimization is performed using Gurobi. The network is bound tightened and added as a constraints to the optimization problem via the Iter-RR algorithm from (Márquez-Neila et al., 2017. The threshold is tuned individually for each scoring system based on the weights selected.

Results: The summary statistics and interpretability assessment for each of the scoring systems can be seen in Table 1. All of the generated scoring systems are sparse, with an average number of features in the system below 3 for all datasets, All scoring systems have integer coefficients below 100, satisfying the intuitiveness criteria for these systems. The consistency of the approach on each of the datasets will be assessed in more detail as the individual scoring systems are examined. Tables 2, 4, 5, 6 show the scoring systems created for the cancer, bank, adult and glaucoma datasets respectively.

Dataset	Average Accuracy	Accuracy Variance	Average Fidelity	Average Sparsity
Cancer	0.82	0.01	0.88	2.7
Bank	0.88	0.002	1	1
Adult	0.76	0.0004	0.99	1.1
Glaucoma	0.625	0.008	0.74	1.6

Table 1: Scoring System Performance Statistics

The summary statistics in Table 1 show that the hypothesis is supported by some of scoring systems produced. All of the scoring systems produced are well within the interpretability requirements, as can seen with inspection of the resulting scoring systems. The accuracy and fidelity results also show some strong results. The scoring systems produced for the the bank and adult datasets show almost perfect fidelity to the original neural networks, which is a success and supports the hypothesis. The cancer datasets fidelity is high but not quite high enough to be considered a success. The variability in the average accuracy of the scoring system shows their are some systems that perform well where others struggled. Finally, the glaucoma dataset is the worst performer, with an unsatisfactory accuracy and fidelity rating. There are two non mutually exclusive causes for the poor accuracy. There could be some nonlinear behavior captured by the neural network not translating by the current approach, and given the sparsity of the resulting scoring systems they could be too sparse.

The ten different scoring systems produced to score the cancer dataset, shown in Table 2, demonstrate the approach failed the consistency criteria. To be confident that the algorithm has produced a representative scoring system, it would expected that several of the networks would produce the same scoring system. There are two possible primary sources of the variation in the scoring system, variations in the learned neural networks and variation from the scoring system generation process. In order to investigate the variation in the neural networks learned, the relevance of each of the inputs is compared across the different neural networks in Table 3.

Scoring System	Threshold	Number of times Generated
Uniformity of Cell Size + 2*Single Epithelial Cell Size + Bare Nuclei + 3*Normal Nucleoli	22	1
Clump Thickness + 10*Single Epithelial Cell Size + 11*Bland Chromatin + 6*Mitosis	85	1
Clump Thickness + Uniformity of Cell Size + 2*Uniformity of Cell Shape + 4*Normal Nucleoli + Mitoses	27	1
Clump Thickness + Uniformity of Cell Shape + Bare Nuclei	11	1
Uniformity of Cell Size + 3*Normal Nucleoli	12	1
Clump Thickness	4	1
Uniformity of Cell Size + Marginal Adhesion	6	1
2*Uniformity of Cell Size + Single Epithelial Cell Size + Bare Nuclei	13	1
Uniformity of Cell Size + Marginal Adhesion	6	1
3*Marginal Adhesion	8	1

Table 2: Scoring Systems Produced for Cancer Dataset

The data contained in Table 3 shows the normalized aggregated relevance scores for all of the input features of the cancer dataset. A higher score means that the input feature was more relevant in aggregate over all test examples to the classification decision, and the entries are shaded by their relevance scores, so that a darker shade is a more relevant entry for that network. The bolded entries represent the features selected to be part of the final scoring system for that fold. The variation in relevant features shows that the variation in the resulting scoring system is influenced at least in part by the different neural networks generated during each of the cross-validation folds. Another interesting observation is how the resulting scoring system does not always include the input features the network found to be the most relevant. In fold 4, the two most relevant inputs were not selected in the final scoring system, however, this scoring system was the best performing of the ten generated and had a perfect accuracy on the testing dataset.

Additionally of note, the scoring system generated in fold 4, which is the best performing of the ten generated by this approach,

$$ClumpThickness + UniformityofCellShape + BareNuclei - 11$$

Clump Thickness	Uniform of Cell Size	Uniform of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses
0.34	**0.79**	0.21	0.25	**0.33**	**0.24**	0.83	**0.12**	0.19
0.32	0.33	0.05	0.1	**0.13**	0.27	**0.36**	0.21	**0.99**
0.20	**0.03**	**0.55**	0.32	0.55	0.47	0.92	**0.22**	**0.35**
0.06	0.07	**0.32**	0.04	0.38	**0.07**	0.12	0.02	0.9
0.14	**0.10**	0.07	0.15	1	0.01	0.44	**0.06**	0.06
NA	NA	NA	NA	NA	NA	NA	NA	NA
0.82	**0.48**	0.28	**0.43**	0.36	0.17	0.93	0.01	0.86
0.49	**0.32**	0.1	0.28	**0.27**	**0.19**	0.52	0.19	0.48
0.06	**0.10**	0.01	**0.16**	0.41	0.09	0.17	0.04	0.92
0.11	0.26	0.1	**0.06**	0.6	0.08	0.24	0.08	0.99

Table 3: Relevance Scores for Cancer Scoring System Features

is almost identical to the one generated by SLIM, another approach to learning scoring systems for classification and published by (Ustun et al.,2013)for the same dataset. As this scoring system is the only one presented for that dataset, the default average performance would exceed that of this method. However, if the best system from this paper is selected, the approaches perform similarly on this dataset.

Bank Scoring System	Threshold	Number of times Generated
Age	41	5
Campaign	3	2
Duration	263	1
Poutcomeother	0	1
June	0	1

Table 4: Scoring Systems Produced for Bank Dataset

The scoring systems generated for the bank dataset are more consistent across the different cross validation folds. Their almost perfect fidelity show that these systems are characterizing the behavior of the neural networks well. The variation in the features selected can partially be explained by overfitting. The poutcomeother and June scoring systems both perform better on the test set than the trained neural network, which is an indication this single variable scoring system may be overfitting. Additionally, the bank dataset is heavily skewed towards the negative class, with 88% of the examples belonging to the negative class. Thus, while the scoring system is a good fit for the neural network, the networks cannot be said to be a good model for the problem.

Adult Scoring System	Threshold	Number of times Generated
Age	39	5
Capital Gain + 4*Hours Per Week	1258	1
Capital Gain	1057	1
Capital Loss	88	1
Female	0	1
Husband	0	1

Table 5: Scoring Systems Produced for Adult Dataset

The scoring systems generated for the adult dataset show similar behavior to the bank dataset. Their almost perfect fidelity show that these systems are characterizing the behavior of the neural networks well. The variation in the features selected can partially be explained by overfitting. The $CaptialGain + 4 * HoursPerWeek$ and $husband$ scoring systems both perform better on the test set than the trained neural network, which is an indication of overfitting. The adult dataset has a similar skew as the bank dataset, with 76% of the examples belonging to the negative class. Thus, while the scoring system is a good fit for the neural network, again the neural network does not model the problem domain well as it fails to predict the positive class. Interestingly, while the overall accuracy between the scoring system is comparable, the scoring systems on average predict the positive class correctly 20% of the time compared to less than 5% from the trained networks.

Glaucoma Scoring System	Threshold	Number of times Generated
clv	23	4
6*abrs + 40*mdg +57*tension + 40*clv	1878	1
tmn	0	1
23*tmn + 87*tension + 70*clv	2994	1
6*clv	135	1
ag + clv	25	1
phct	0	1

Table 6: Scoring Systems Produced for Glaucoma Dataset

While the fidelity of the scoring systems produced for the Glaucoma dataset is the smallest of the four datasets considered, it shows more consistency in the features selected than the cancer dataset. Part of the improved consistency is the influence of the relevance aggregation criteria on the construction of the scoring systems. The impact of relevance aggregation constraint included in the approach is most notable in the results of the glaucoma dataset. The idea behind the constraint was to provide additional sparsity by constraining features determined to be unrelated to the problem. The impact is not seen in the other three dataset results as the cancer datasets features are already sparse, and the bank and adult datasets are represented well by single variables. All of those relevance thresholds were tuned to 0.05 in the case of the cancer dataset or 0 in the case of the bank and adult datasets. However, increasing the relevance threshold for the Glaucoma from 0.05 to 0.25 increased the average accuracy by 5% and increased the average sparsity of the resulting scoring systems from 5.3 to 1.6. Increasing the relevance further began to decrease the accuracy of the scoring systems. 7 shows the relevance scores for the variables in the dataset that were included in a final scoring system. In this instance, the relevance is closely related to which variables occur how frequently in the scoring system.

ag	abrs	phct	mdg	tmn	tension	clv
0.087	0.072	0.13	0.13	0.12	0.45	**0.83**
0.10	0.10	0.092	0.063	0.055	0.18	**0.99**
0.18	**0.26**	0.084	**0.26**	0.12	**0.51**	**0.65**
0.10	0.045	0.067	0.16	**0.29**	0.60	0.69
0.12	0.12	0.016	0.097	**0.27**	**0.33**	**0.99**
0.09	0.05	0.014	0.016	0.017	0.2465921	**1**
0.32	0.08	0.18	0.18	0.095	0.16	**0.80**
0.076	0.032	0.072	0.023	0.048	0.46	**0.91**
0.10	0.16	**0.32**	0.11	0.17	1	0.13
0.46	0.13	0.24	0.15	0.053	0.42	**0.74**

Table 7: Relevance Scores for Glaucoma Scoring System Features

5. Conclusion

This paper demonstrates a method for learning scoring systems from a deep rectifier (ReLu) neural network. In all examples, an easily interpretable scoring system is generated with varying degrees of fidelity to the original model. Unfortunately, the datasets where the scoring systems exhibit the highest fidelity to the neural networks behavior are the ones where the neural networks has overfit the problem and is not a good representation of the dataset. The wider benefit of this approach would be to create high fidelity scoring systems from problems where deep neural networks have a strong inductive bias and outperform other methods. For the two datasets with high classification accuracy that are closest to that goal, the fidelity of the average scoring system to the original network does not yet meet that goal.

Further work should focus on improving the accuracy of the scoring systems. While the scoring systems produced currently are sparse, removing some of the sparsity in exchange for accuracy would be a net improvement. Demonstrating a high fidelity scoring system from a highly accurate deep rectifier neural network is the next step for this research. Furthermore, all of the problems examined in this piece are binary classification problems. How to extend the approach to multiclass classification problems is another area for further research.

References

Sercan O Arık and Tomas Pfister. Protoattend: Attention-based prototypical learning. *Journal of Machine Learning Research*, 21:1–35, 2020.

Messod D Beneish, Charles MC Lee, and D Craig Nichols. Earnings manipulation and expected returns. *Financial Analysts Journal*, 69(2):57–82, 2013.

Emilio Carrizosa, Amaya Nogales-Gómez, and Dolores Romero Morales. Strongly agree or strongly disagree?: Rating features in support vector machines. *Information Sciences*, 329:256–273, 2016.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

FICO. Introduction to scorecard for fico model builder, 2011. URL `http://www.fico.com/en/node/8140?file=7900`.

Brian F Gage, Amy D Waterman, William Shannon, Michael Boechler, Michael W Rich, and Martha J Radford. Validation of clinical classification schemes for predicting stroke: results from the national registry of atrial fibrillation. *Jama*, 285(22):2864–2870, 2001.

Bruno Iochins Grisci, Mathias J Krause, and Marcio Dorn. Relevance aggregation for neural networks interpretability and knowledge discovery on tabular data. *Information Sciences*, 559:111–129, 2021.

Edward Latessa, Paula Smith, Richard Lemke, Matthew Makarios, and Christopher Lowenkamp. Creation and validation of the ohio risk assessment system: Final report. *Center for Criminal Justice Research, School of Criminal Justice, University of Cincinnati, Cincinnati, OH. Retrieved from http://www. ocjs. ohio. gov/ORAS_FinalReport. pd f*, 2009.

O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. *SIAM News, Volume 23, Number 5,pp 1-18*, September 1990.

Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Imposing hard constraints on deep networks: Promises and limitations. *arXiv preprint arXiv:1706.02025*, 2017.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. *arXiv preprint arXiv:2010.09337*, 2020.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. Springer International Publishing, Cham, 2019. ISBN 978-3-030-28954-6. doi: 10.1007/978-3-030-28954-6_10. URL `https://doi.org/10.1007/978-3-030-28954-6_10`.

Rui P Moreno, Philipp GH Metnitz, Eduardo Almeida, Barbara Jordan, Peter Bauer, Ricardo Abizanda Campos, Gaetano Iapichino, David Edbrooke, Maurizia Capuzzo, and Jean-Roger Le Gall. Saps 3—from evaluation of the patient to evaluation of the intensive care unit. part 2: Development of a prognostic model for hospital mortality at icu admission. *Intensive care medicine*, 31(10):1345–1355, 2005.

Sergio Moro, Raul Laureano, and Paulo Cortez. Using data mining for bank direct marketing: An application of the crisp-dm methodology. 2011.

Nataliya Sokolovska, Yann Chevaleyre, and Jean-Daniel Zucker. A provable algorithm for learning interpretable scoring systems. In *International Conference on Artificial Intelligence and Statistics*, pages 566–574. PMLR, 2018.

Christopher D Still, G Craig Wood, Peter Benotti, Anthony T Petrick, Jon Gabrielsen, William E Strodel, Anna Ibele, Jamie Seiler, Brian A Irving, Melisa P Celaya, et al. A probability score for preoperative prediction of type 2 diabetes remission following rygb surgery. *The lancet. Diabetes & endocrinology*, 2(1):38, 2014.

Berk Ustun and Cynthia Rudin. Learning optimized risk scores. *Journal of Machine Learning Research*, 20(150):1–75, 2019.

Berk Ustun, Stefano Traca, and Cynthia Rudin. Supersparse linear integer models for interpretable classification. *arXiv preprint arXiv:1306.6677*, 2013.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.