PLAY: A Profiled Linear Weighting Scheme for Understanding the Influence of Input Variables on the Output of a Deep Artificial Neural Network

Torsten Dietl, Gjergji Kasneci, Johannes Fürnkranz and Eneldo Loza Mencía

Abstract Recently, deep artificial neural networks (DANNs) have been successfully applied to various pattern recognition tasks with high industrial impact. Their results are so convincing that neural nets are already tested in heavily regulated fields like medicine or finance. However, these autonomous systems are often deployed without evaluating the reasoning behind their decisions. Thus, recent research has shifted towards methods that increase the interpretability of DANNs. The goal of this paper is to explain the influence of input variables on the decision of a DANN. More precisely, we aim at improving the linear weighting scheme for the contribution of input variables (LICON), a

Torsten Dietl
SCHUFA Holding AG, Komoranweg 5, 65201 Wiesbaden
Technische Universität Darmstadt, Karolinenplatz 5, 64289 Darmstadt
✉ Torsten.Dietl@schufa.de

Gjergji Kasneci
SCHUFA Holding AG, Komoranweg 5, 65201 Wiesbaden
✉ Gjergji.Kasneci@schufa.de

Johannes Fürnkranz
Computational Data Analytics Group, Johannes Kepler University, 4040 Linz, Austria
✉ johannes.fuernkranz@jku.at

Eneldo Loza Mencía
Knowledge Engineering Group, TU Darmstadt, Hochschulstraße 10, 64289 Darmstadt
✉ research@eneldo.net

ARCHIVES OF DATA SCIENCE, SERIES A
(ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. 6, No. 1, 2020

DOI: 10.5445/KSP/1000098011/11
ISSN 2363-9881

previously introduced method which estimates the contributions of inputs in a local neighborhood, by combining it with the gobal sensitivity approach(GSA), which uses sampling to examine multiple values of an input. This allows the local influence estimation of LICON to be assessed in relation to estimates obtained from sampled input values. The effectiveness of the proposed approach is assessed via a comparative study of the involved explanation methods. Despite the computational complexity, which has to be dealt with in the future, it is shown that the proposed approach generates reasonable estimates for input contributions.

1 Introduction

Machine learning (ML) currently has a high industrial impact, due to the enormous success in applying deep artificial neural networks (DANNs) to various pattern-recognition tasks (Goodfellow et al, 2016). However, it is inherently difficult to explain the reasoning behind a network's decision. In confronting problems with many input variables and a complex neural network structure, we need to develop methods for understanding the influence of individual input variables on the decision of the neural net.

Thus, recent research was devoted to increase the interpretability of DANNs (Papadokonstantakis et al, 2006; Lin and Cunningham, 1995; Benitez et al, 1997; Green et al, 2009; Gevrey et al, 2003; Smilkov et al, 2017; Baehrens et al, 2010). Maybe the best-known system is the use of locally interpretable, model-agnostic explanations (LIME, Ribeiro et al, 2016) which solve this problem by learning local models that explain the predictions of an opaque model in the neighborhood of a query point. For recent surveys of this quickly developing research area, we refer to (Jair Escalante et al, 2018; Molnar, 2019).

A promising approach is the *linear weighting scheme for the contribution of input variables* (LICON) method by Kasneci and Gottron (2016) which is able to provide reasonable influences of an input on an output for a given example. Furthermore, LICON is able to calculate general influence values for the complete model which gives at least an indication of the overall influence of an input variable. However, LICON suffers from a locality issue, as it uses linear approximations to calculate the influence values. Linear approximations are only reliable for a local interval around the examined input. This work tackles the locality issue by combining the LICON approach with a sampling or profiling concept introduced by Cortez and Embrechts (2011) in their *global sensitivity approach* (GSA), and thereby contributes to the current research regarding the explanation of the decision of a DANN.

The remainder of this paper is organized as follows: Subsequently to this Section, a short overview of the related methods LICON and GSA is given. Thereafter, the new method, a combination of the two approaches, is presented in detail in Section 3. Section 4 describes the general experimental setup and discusses the evaluation results, before Section 5 concludes the paper.

2 Interpreting Deep Networks

As mentioned before, there are multiple ways to analyze the decision of DANNs. The most intuitive method is the *neural interpretation diagram* which displays the architecture of the net as a directed graph and the weights as the thickness of the connection lines. Another way to explain the decision of a DANN is the extraction of rule sets (Andrews et al, 1995; Zilke et al, 2016; González et al, 2017). However, as complex DANNs tend to generate huge rule sets, whose comprehensibility is questionable, this paper focused on approaches that try to calculate aggregated influence values for the input variables. *Connection weight* methods, based on *Garson's algorithm*, were early approaches which calculated aggregated influence values (Garson, 1991; Milne, 1995; Gevrey et al, 2003). However, they only work with three-layer networks. More promising approaches are *sensitivity analysis* and *back-propagation-based* methods (Gevrey et al, 2003; Lek et al, 1996; Reddy et al, 2015; Baehrens et al, 2010; Kasneci and Gottron, 2016; Smilkov et al, 2017; Kindermans et al, 2018; Springenberg et al, 2015; Bach et al, 2015; Montavon et al, 2017). Two of them, the LICON method by Kasneci and Gottron (2016) and the GSA approach by Cortez and Embrechts (2011) are combined for the method proposed in this paper and thus will be described in the following.

2.1 LICON

LICON approximates the function represented by a DANN through the aggregation of local behaviors of its neurons. For the explanation of the local behavior of a neuron, Kasneci and Gottron (2016) used gradients since the usual design of neural networks (using backpropagation as a training method) involve derivable activation functions.

LICON starts by initialising the influence value for every input variable to 1. Then, for each neuron in a layer, it calculates the weighted gradients of the activation function. To aggreate the influence values through out the net, LICON sums up the product of the weighted gradients and the calculated influence values of the previous layer. Thus, we eventually receive a network-wide influence value for every input on each output.

2.2 GSA

Cortez and Embrechts (2011) have developed the global sensitivity approach (GSA) as a generalization of the works of Kewley et al (2000) and Embrechts et al (2003). The GSA method is a sensitivity analysis using profiled inputs to measure the influence of an input variable or a combination of input variables. This means that the method generates input samples based on a base input vector, predicts outputs for these input samples and aggregates them into an influence value using a sensitivity measure.

The algorithm implemented for this research differs slightly from the algorithm described by Cortez and Embrechts (2011) in two aspects. Firstly, to improve comparability between LICON, GSA and the proposed method, a given input example was used as the base input vector instead of an input vector consisting of the mean or median values for every input variable, and the gradient metric was used as a sensitivity measure. Secondly, in analogy to the LICON method, the calculated influence value for the non-target class was interpreted as a negative influence for the input variable on the target class. This was possible because our work is limited to binary classifiers.

3 The Profiled LICON Analysis Approach

In the PLAY method, as seen in Figure 1, a number p of input samples X are created based on a base input vector $x^{(0)}$ using the GSA approach. For this research, samples were generated for each input variable individually.

Afterwards, LICON is applied to every input sample generated in the profiling process. The subsequently produced LICON influences A_j are used to calculate the mean influence value μ_i and the variance σ_i over all samples A_{ji} for an input variable.

```
function $\mathrm{PLAY}(p, x^{(0)})$
    $X \leftarrow GSA.CreateProfiledInputs(p, x^{(0)})$
    for $(x_j \in X)$ do
        $A_j \leftarrow LICON(x_j)$
    end for
    for $i = 0 \ldots |I|$ do
        $\mu_i \leftarrow \frac{1}{|X|} \sum_{j=0}^{|X|} A_{ji}$
        $\sigma_i \leftarrow \frac{1}{|X|} \sum_{j=0}^{|X|} (A_{ji} - \mu_i)^2$
    end for
    return $\{\mu, \sigma\}$
end function
```

Figure 1: PLAY algorithm as implemented for this resarch.

As in LICON, only the influence values for the target class output are used in the evaluation of the resulting data because we restrict ourselves to binary classification. Furthermore, all influence values of discrete input variables are aggregated by separately adding up positive and negative influences of a discrete input variable on the target class.

4 Experimental Evaluation

To show the validity of the proposed PLAY method, we performed the following experiments to compare the PLAY method to its components, LICON and GSA.

4.1 General Experimental Setup

The experiments were limited to feed-forward neural networks containing a softmax layer with two neurons as the output layer [1]. However, restricting the experiments to binary classifiers was no limitation in the application of the methods since each multi-class problem can be solved by multiple binary classifiers working in a one-versus-all manner. Additionally, discrete input variables were one-hot-encoded.

In the experiments, three DANNs were trained on the following data sets: An artificial data set (laboratory condition), the German credit data set (real-world example) and the MNIST database of handwritten digits (easily interpretable benchmark) (Dua and Karra Taniskidou, 2017; LeCun et al, 1998). The architecture of the DANNs was chosen previously by 10-fold cross validation. Subsequently to the training, each method was applied on chosen inputs of the training data set. For the GSA and PLAY methods, six input samples for every input feature were created by generating equally distributed profile points over the complete input interval, following the approach by Kewley et al (2000). In this way, every variable was profiled independently while the other variables were fixed to their value in the base input vector. The resulting influences were then assessed to identify the methods which calculate the influences most accurately.

In a final step, the discriminatory power of the methods was evaluated, i.e. the most *selective* method was determined by comparing weighted AUC and GINI[2] coefficient values of a 10-fold cross validation. The cross validation was performed with logistic regression models trained on the produced influence values (positive, negative, mean, and variance) for every input variable.

4.2 Artificial Data Set

The five input features of the 1000 instances for the artificial data set were sampled from the Gaussian distribution. We ensured the correlations (0.8, 0.6,

[1] The used nodes in each hidden layers were: Artificial: one layer, 5 logistic neurons; German credit: two layers, 33 logistic and 23 rectifier neurons; MNIST: three layers, 33, 13 and 2 logistic neurons.

[2] The GINI coefficient is defined as $GINI = 2 \cdot AUC - 1$ (Hand and Till, 2001).

0.4, 0.2, and 0.0) between the inputs and the output by the following process: The probability of sampling the positive class depended linearly on the input values and the given correlations as coefficients.

This was used to assess how far the calculated input influences reproduce *real* correlations. As the artificial data set provides a given correlation between the input variables and the output, it was easy to determine the method which calculated the influences the most accurately.

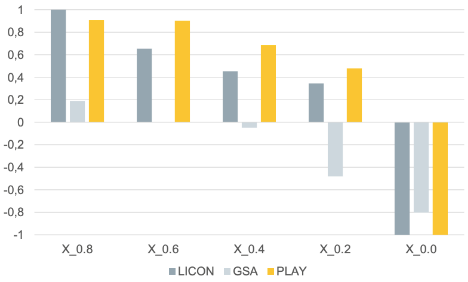

Figure 2: Mean influence values for all three methods (LICON, GSA, PLAY) applied to the artificial data set.

As seen in Figure 2, the LICON method delivers a good approximation and correctly identifies the order of the influences. Furthermore, it exhibits a low variance for the inputs with a given correlation. For the input variable $X_{0.8}$ the variance is 0, and for the variables $x_{0.6}$, $x_{0.4}$ and $x_{0.2}$ the variance is near 0.15. However, the influence of the non-correlated input variable $x_{0.0}$ is calculated incorrectly and has a high variance of 10.676. This high variance implies that it was not possible for LICON to accurately identify the non-correlated variable.

Similar to LICON, the GSA method identified the same order between the input variables. However, the influence values differ considerably from those calculated by the LICON method and from the expected values, as seen in Figure 2. This poor performance of the GSA method is further confirmed by the relatively high variance values (around 30). Thus, GSA was not able to calculate the correct influence values for the artificial data set.

Finally, Figure 2 shows that PLAY calculated influences for the artificial data set more accurately than GSA, but less so than LICON. Although PLAY made the same mistake as LICON by overestimating the influence of the uncorrelated

variable x_0, it was nevertheless able to return the correct order of influence values ($x_{0.8} = 0.908$, $x_{0.6} = 0.903$, $x_{0.4} = 0.685$, $x_{0.2} = 0.479$ and $x_{0.0} = -1$). However, it also overestimated the influence of the input variables $x_{0.6}$, $x_{0.4}$ and $x_{0.2}$. Due to the relatively high variances (approx. 25), the accuracy of the calculated influence values could thus be questioned.
As seen in Figure 3, the PLAY approach achieved a higher AUC and GINI value for the artificial data set than LICON. This indicates that PLAY is more selective in this case, and it suggests that the combination of LICON with the profile approach of the GSA method is useful.

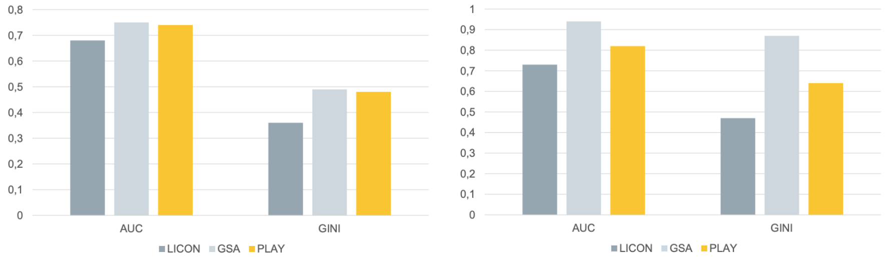

Figure 3: Selectivity measures (AUC and GINI) for all three methods (LICON, GSA, PLAY) applied to the artificial data set (left) and the german credit data set (right).

4.3 German Credit Data Set

To cover the domain of real-world data, the German credit data set was selected (Dua and Karra Taniskidou, 2017). The data set was chosen because it mimics the data of a heavily regulated field which requires reliable explanation methods to be able to apply DANN in the credit decision process. It consists of 20 input features (categorical and numerical data) and one binary output.

To assess the most accurate method, the calculated influences were compared to the correlation coefficients of a linear regression applied on the German credit data set. This was done because the German credit data set can be solved by linear regression which is a commonly used and accepted method in practice (van der Ploeg, 2010) and because no predefined correlation between the variables existed.

As seen in Figure 4, the LICON method correctly identified the variables *duration*, *credit amount*, *instalment commitment*, *personal status* and *existing credits* as negative influences. Furthermore, it correctly identified the variables *saving status*, *job* and *foreign worker* as positive influences. However, besides the correctly identified variables, LICON produced some reasonable differences when compared to the expected values. For example, the variable *credit history* did not only contain unpaid or delayed credit rates but also credits that were paid on time. Hence, it makes sense that a consumer who does pay their credit rates on time is considered a low risk and ,therefore, a good consumer. Furthermore, if a customer has payment plans at other banks, it seems reasonable to expect a basic trust in the reliability of the customer, because other banks trust the customer.

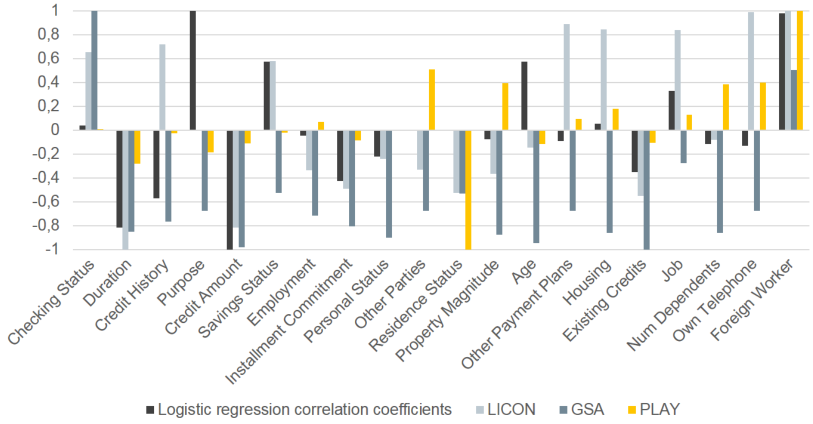

Figure 4: Correlation coefficients of the linear regression (expected values) and mean influence values for all three methods (LICON, GSA, PLAY) applied to the German credit data set.

The GSA method performed substantially worse than LICON. It essentially estimated almost all influences to be negative. This neither fits to the expected values nor do we have a reasonable explanation for this behavior.

In comparison, the PLAY method provides reasonable influence values. For example, the variables *other parties* and *property magnitude*, whose influence values differ from the expected values and the values calculated by the LICON method, remain intuitively correct. If two people apply for a credit instead of only one person (other parties), the risk for the bank is lowered. Furthermore, the risk is lowered if a customer can provide a property with an equivalent value as a safety (property magnitude). Finally, it is interesting to consider that the influence value for the variable *purpose* produced by PLAY differs from the influence value produced by LICON, even though PLAY is based on LICON. This difference, as well as the higher variances of the influence values calculated by the PLAY method, could be caused by the profiling process and could, therefore, indicate that PLAY is able to capture the relation between the individual manifestations of the input variables.

For the German credit data set PLAY achieved higher AUC and GINI values than LICON, as can be seen in Figure 3. This indicates that the PLAY method is more selective, in this case, and confirms that the combination of LICON with the profile approach of GSA is useful.

4.4 MNIST Database of Handwritten Digits

As a final benchmark, we selected the MNIST database of handwritten digits (LeCun et al, 1998) which is a commonly accepted and frequently used dataset in the research field of DANNs (Keysers, 2007; Mizukami et al, 2010). The classification task is easily solvable for humans, and, thus, the calculated influences can be assessed for being plausible by visual inspection. Therefore, following Kasneci and Gottron (2016), the resulting influences were displayed as images for visual inspection. In the resulting images, positive or supporting influences are colored in blue, while negative or rejecting influences are displayed in red. Furthermore, the original input, for which the influence values were calculated, is projected on the influence image. In doing so, it is possible to examine, if the pixel expected to support/reject the classification match the calculated supporting/rejecting pixel. Thus, it is possible to compare the calculated influence values against the human intuition.

Table 1: Custom illustration of advantages and disadvantages of the three methods in comparison.

	LICON	**GSA**	**PLAY**
Complexity of the algorithm	$O(L \cdot N^3)$	$O(P \cdot L \cdot N^3)$	$O(P \cdot L \cdot N^4)$
Intuition of explanation (based on MNIST)	5/5	0/5	5/5
Accuracy of explanation	4/5	0/5	3/5
Information content of explanation	3/5	–*	4/5

L Number of layers.

N (Maximum) number of neurons for each layer (including input neurons).

P Number of profile/sample points.

$x/5$ Rating of the method (0 = not intuitive / accurate / informative; 5 = perfectly intuitve / accurate / informative).

$*$ Not applicable, because the results of the method were not accurate.

Due to the high sampling costs (as seen in Table 1) because of the relatively high number of input variables (compared to the previous data sets), we restricted our analysis to 100 records per digit-specific data set. To obtain a well-balanced data set for the evaluation, the first 50 records of the target class and the first 50 records of the non-target class were taken. The selected records were then shuffled to prevent any bias or semi-optimal solution for the selectivity test.

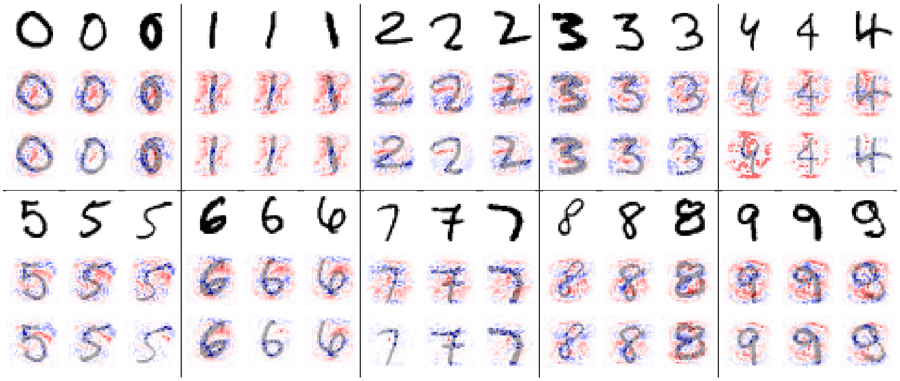

Figure 5: Influence values for each digit from 0 to 9 (first row) calculated by the LICON (second row) and the PLAY (third row) method on the MNIST database. Shown are three examples for each digit.

Figure 5 shows the influences estimated by they PLAY method compared to the influences calculated by the LICON approach. They are similar to those presented by Kasneci and Gottron (2016). A notable difference is, however, a greater distinction between the individual positive and negative influence areas. The blue and red pixels in the images created by the LICON method often join together, with only a few white pixels in between. In comparison, pixels in the images produced by the PLAY method are more concentrated and thus provide a clearer picture of the significant influence areas.

Furthermore, similar to the selectivity tests performed with the influence data sets that were calculated from the artificial and the German credit data sets, a selectivity test was performed for the influence data sets produced from the digit-specific MNIST data sets. However, the results were unsatisfactory because the digit-specific MNIST data sets contained only 100 records each, whereas the logistic regression model had a degree of freedom of 3,137; allowing it to *remember* the class values for every record. Thus, all of the performed logistic regressions achieved AUC and GINI values of ~ 1. As a consequence it was not possible to draw any useful or reliable conclusions from this part of the experiment.

5 Conclusion and Outlook

The overall aim of this paper was to improve the LICON method, tackle its locality issue and thereby contribute to the research regarding the explanation of the influence of input variables on the decision of a DANN. Based on three different experiments it was shown that PLAY calculates reasonable and understandable influence values. Additionally, PLAY achieved higher selectivity measures compared to the LICON method. Furthermore, the profiling approach used in the PLAY method allows for a closer look into the influences of an input variable and provides a more detailed trend of the influence value.

Even though the PLAY method presented itself with overall good results in this research, a few points for possible improvement have to be mentioned. First of all, some of the evaluated results showed that the PLAY method is imprecise in certain cases (overestimation of variables). Another limitation of PLAY is its computational cost. Therefore, its use can currently only be recommended if applied as an in-depth examination of a single classification example. For a time-efficient overview of the influences for a complete data set, this research revealed that, as of now, the LICON method should be prefered. Finally, this research was not able to test the PLAY method with a combination of inputs or for other domains. Thus, future research will try to reduce the computational complexity, as well as look into input combinations and data sets of other domains.

References

Andrews R, Diederich J, Tickle AB (1995) Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based Systems 8(6):373–389. DOI: 10.1016/0950-7051(96)81920-4.

Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W (2015) On Pixel-wise Explanations for Non-Linear Classifier Decisions by Layer-wise Relevance Propagation. PLoS ONE 10(7):1–44. DOI: 10.1371/journal.pone.0130140.

Baehrens D, Schroeter T, Harmeling S, Kawanabe M, Hansen K, Müller KR (2010) How to Explain Individual Classification Decisions. Journal of Machine Learning Research 11(Jun):1803–1831. URL: http://jmlr.org/papers/v11/baehrens10a.html.

Benitez JM, Castro JL, Requena I (1997) Are Artificial Neural networks Black Boxes? IEEE Transactions on Neural Networks 8(5):1156–1164. DOI: 10.1109/72.623216.

Cortez P, Embrechts MJ (2011) Opening Black Box Data Mining Models Using Sensitivity Analysis. In: Proc. IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 341–348. DOI: 10.1109/CIDM.2011.5949423.

Dua D, Karra Taniskidou E (2017) UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. URL: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data). Online. Accessed 16 September 2018.

Embrechts M, Ozdemir FAM, Kewley R (2003) Data mining for molecules with 2-D neural network sensitivity analysis. International Journal of Smart Engineering System Design 5(4):225–239. DOI: 10.1080/10255810390245555.

Garson GD (1991) Interpreting neural network connection weights. Artificial Intelligence Expert 6:47–51. DOI: 10.21037/atm.2018.05.32.

Gevrey M, Dimopoulos I, Lek S (2003) Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling 160:249–264. DOI: 10.1016/S0304-3800(02)00257-0.

González C, Loza Mencía E, Fürnkranz J (2017) Re-training Deep Neural Networks to Facilitate Boolean Concept Extraction. In: Discovery Science, Yamamoto A, Kida T, Uno T, Kuboyama T (eds), Springer International Publishing, Cham, pp. 127–143. DOI: 10.1007/978-3-319-67786-6_10.

Goodfellow IJ, Bengio Y, Courville AC (2016) Deep Learning. Adaptive computation and machine learning, MIT Press. URL: http://www.deeplearningbook.org/.

Green M, Ekelund U, Edenbrandt L, Björk J, Forberg JL, Ohlsson M (2009) Exploring new possibilities for case-based explanation of artificial neural network ensembles. Neural Networks 22(1):75–81. DOI: 10.1016/j.neunet.2008.09.014.

Hand DJ, Till RJ (2001) A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Machine Learning 45(2):171–186. DOI: 10.1023/A:1010920819831.

Jair Escalante H, Escalera S, Guyon I, Baró X, Güçlütürk Y, Güçlü U, van Gerven MAJ (2018) Explainable and Interpretable Models in Computer Vision and Machine Learning. The Springer Series on Challenges in Machine Learning, Springer-Verlag. DOI: 10.1007/978-3-319-98131-4_4.

Kasneci G, Gottron T (2016) LICON: A Linear Weighting Scheme for the Contribution of Input Variables in Deep Artificial Neural Networks. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM), Mukhopadhyay S, Zhai C (eds), ACM, New York, NY, USA, no. 10 in CIKM 2016, pp. 45–54. DOI: 10.1145/2983323.2983746.

Kewley RH, Embrechts MJ, Breneman C (2000) Data strip mining for the virtual design of pharmaceuticals with neural networks. IEEE Transactions on Neural Networks 11(3):668–679. DOI: 10.1109/72.846738.

Keysers D (2007) Comparison and Combination of State-of-the-art Techniques for Handwritten Character Recognition: Topping the MNIST Benchmark. ArXiv e-prints. 0710.2231.

Kindermans P, Schütt KT, Alber M, Müller K, Erhan D, Kim B, Dähne S (2018) Learning how to explain neural networks: PatternNet and PatternAttribution. In: Proc. 6th International Conference on Learning Representations, (ICLR) 2018, Vancouver, BC, Canada. URL: `https://openreview.net/forum?id=Hkn7CBaTW`.

LeCun Y, Cortes C, Burges CJ (1998) The MNIST database of handwritten digits. URL: `http://yann.lecun.com/exdb/mnist/`. Online. Accessed 16 September 2018.

Lek S, Belaud A, Baran P, Dimopoulos I, Delacoste M (1996) Role of some environmental variables in trout abundance models using neural networks. Aquatic Living Resources 9(1):23–29. DOI: 10.1051/alr:1996004.

Lin Y, Cunningham GA (1995) A new Approach to Fuzzy-Neural System Modeling. IEEE Transactions on Fuzzy Systems 3(2):190–198. DOI: 10.1109/91.388173.

Milne L (1995) Feature Selection Using Neural Networks with Contribution Measures. In: Australian Conference on Artificial Intelligence AI'95, Yao X (ed), Canberra, Australia, Vol. 9, pp. 124–136. DOI: 10.1051/alr:1996004.

Mizukami Y, Tadamura K, Warrell J, Li P, Prince S (2010) CUDA Implementation of Deformable Pattern Recognition and its Application to MNIST Handwritten Digit Database. In: Proc. 20th International Conference on Pattern Recognition, IEEE, pp. 2001–2004. DOI: 10.1109/ICPR.2010.493.

Molnar C (2019) Interpretable Machine Learning – A Guide for Making Black Box Models Explainable. URL: `https://christophm.github.io/interpretable-ml-book/`.

Montavon G, Lapuschkin S, Binder A, Samek W, Müller KR (2017) Explaining nonlinear classification decisions with deep Taylor decomposition. Pattern Recognition 65(11):211–222. DOI: 10.1016/j.patcog.2016.11.008.

Papadokonstantakis S, Lygeros A, Jacobsson SP (2006) Comparison of recent methods for inference of variable influence in neural networks. Neural Networks 19(4):500–513. DOI: 10.1016/j.neunet.2005.09.002.

van der Ploeg S (2010) Bank Default Prediction Models: A Comparison and an Application to Credit Rating Transitions. URL: `https://thesis.eur.nl/pub/6470/294726ploegma0110.pdf`. Master Thesis.

Reddy NS, Panigrahi BB, Ho CM, Kim JH, Lee CS (2015) Artificial neural network modeling on the relative importance of alloying elements and heat treatment temperature to the stability of α and β phase in titanium alloys. Computational Materials Science 107:175–183. DOI: 10.1016/j.commatsci.2015.05.026.

Ribeiro MT, Singh S, Guestrin C (2016) "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-16), Krishnapuram B, Shah M, Smola AJ, Aggarwal CC, Shen D, Rastogi R (eds), ACM, San Francisco, CA, USA, pp. 1135–1144. DOI: 10.18653/v1/N16-3020.

Smilkov D, Thorat N, Kim B, Viegas F, Wattenberg M (2017) SmoothGrad: removing noise by adding noise. ArXiv e-prints. 1706.03825v1.

Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2015) Striving for Simplicity: The All Convolutional Net. ArXiv e-prints. 1412.6806v3.

Zilke JR, Loza Mencía E, Janssen F (2016) DeepRED - Rule Extraction from Deep Neural Networks. In: Proc. Discovery Science, Calders T, Ceci M, Malerba D (eds), Springer, Cham, pp. 457–473. DOI: 10.1007/978-3-319-46307-0_29.