arXiv:1904.09615v1 [cs.LG] 21 Apr 2019

# Explaining a prediction in some nonlinear models

Cosimo Izzo

*University College London, Gower St, London WC1E 6BT, UK*

## Abstract

In this article we will analyse how to compute the contribution of each input value to its aggregate output in some nonlinear models. Regression and classification applications, together with related algorithms for deep neural networks are presented. The proposed approach merges two methods currently present in the literature: integrated gradient and deep Taylor decomposition.

*Keywords:* Nonlinear models, Deep neural networks, Classification, Regression, Output explanation, Taylor decomposition

## 1. Introduction and Motivation

Nonlinear models have been widely used especially for classification problems in many fields. Nevertheless, one of the main issues encountered by practitioners and academics is strictly related to the interpretation and explanation[1] of the models' output with respect to the input data.

Indeed, focus on the issue dates back in the past also with respect to simple classifiers such as logit and probit. At the best of our knowledge, the first to approach output explanation with respect to some input data in non linear models using a stepwise Taylor decomposition were [5]. They proposed a methodology to explain a change in a left hand side variable due to changes in right hand side variables. Further, they apply the method to presidential voting using the 1972-76 National Election Studies (NES) panel. Among the results in their paper, they conclude that, in logit models, a second order Taylor expansion[2] is found to be enough for a $\Delta x_j <= 0.5/b_j$; while, for higher differences among the right hand side variables, a stepwise method is advised by the authors.

Successively, [3] provided theoretical foundation for using the Taylor decomposition to extend the Oaxaca method to nonlinear functions.

Nowadays, with a growing interest in machine learning tools from many fields, decision makers have sometime shown reluctance in those models because perceived as black boxes. Indeed, in disciplines such as economics, finance and healthcare, explaining predictions is as important as having a well performing model [17, 7, 4].

More in general, a graphical representation of model's choices can help debugging it.

Therefore, many scientists have dedicated their research focuses on trying to shed lights on decisions made by those models. A recent survey of the literature is provided by [8].

Overall, methods present in the literature to explain models'decisions could be distinguished in approaches involving auxiliary models fitting, and other solving the problem directly.

Usually, the former provide less accuracy but more flexibility and therefore sometimes referred as agnostic[3]; while, the latter more accuracy but at the cost of being applicable only to certain class of models.

In this work, we will present a method that is applicable only to a given class of models, but on the other side has maximal accuracy. Indeed, it is the researcher to define the approximation error. Moreover, the method is suitable both for classification and regression.

The algorithm proposed in this article merges two works: the integrated gradient (IG) of [15] and the deep

*Email address:* cosimo.izzo.18@ucl.ac.uk (Cosimo Izzo)

[1] Here for interpretation and explanation we mean the mapping of abstract concepts to the human world, and their graphical representation to explain a decision made by the model [11].

[2] Formally, they drop crossproduct terms from the second order expansion.

[3] Being not tied to a particular type of black box. Among the agnostic methods, the LIME technique from [13] is perhaps the most popular.

Taylor decomposition (DTD) of [10].

Regarding the DTD, it can be considered as an evolution of the layer-wise relevance propagation, technique first adopted in the pixel-wise decomposition method of [2]. The DTD consists in applying a Taylor decomposition in each neuron of the neural network, this allows a back propagation of the output towards its inputs.

A drawback of this methodology stands in the fact that the link function is not always well approximated by a one step first order Taylor decomposition [4].

On the other side, the IG approach could be interpreted as a generalization of the method proposed by [5] in logit models. It consists in carrying out a stepwise first order Taylor expansion of the neural network, where the initial root vector (so called baseline) could be found by imposing that the total output evaluated at this point is equal to zero.

As will be clearer in what follows, finding this baseline with respect to the full model is not always straightforward (especially when a theoretical baseline is not available). Nevertheless, it is trivial when the model is monotonic in each of its inputs. More so, there are cases where the output of the model is not in the set [0, 1], but for examples in all $\Re$[5].

These issues have motivated our work, and the proposal of a novel algorithm that combines the IG and the DTD. The rest of the article is organized as follows. In section 2, we present theoretical support for our method, together with the algorithms that implement it; a comparison with related works will be done in this section as well. Section 3 is dedicated to empirical applications, both to a regression problem and to a classification one. Finally, section 4 concludes.

## 2. Explaining a prediction

When presenting the algorithms, we will be dealing with already estimated models. Therefore, we are not concerned with estimation. On the other side, we will be analysing explanation of a given conditional output with respect to its conditioning inputs, and given the optimal parametrization. Additionally, and without loss of generality, inputs are assumed to be scaled.

### 2.1. Simple input-output models

In this section we propose a variation of the approach used by [5] for logit models. The following Claim can be considered a reformulation of the fundamental theorem of calculus for path integrals (Proposition 1 in [15]).

**Claim 1.** *Given a model of the form $y_t = G(y_t^*) = G(\beta' * x_t)$, with $G(.)$ being differentiable and monotonic in each of its inputs and with image in $[a, b]$, $\beta \in \Re_+^K \setminus \{+\inf\}$ without loss of generality, and $x_t \in X$ bounded subset of $\Re^K$, and $y_t^* \in Y$ bounded subset of $\Re$, then approximate contribution of input $x_{t,k}$ is:*

$$\rho_{t,k} = \sum_{y^*=\beta' * x_0}^{y_t^*=\beta' * x_t} g(y^*) * \beta_k * \Delta x_k \tag{1}$$

*Where $x_0$ is the starting root point, i.e.: $x_{i,0} = c <= min(X)$ $\forall i = 1, ..., K$ and $G(\beta' * x_0) = a$.*

The Claim can be easily proven by: exploiting the monotonic property of $G(.)$, using the definition of integral and that of derivative, applying Taylor decomposition. Namely,

$$G(y_t^*) = \int_{-\infty}^{y_t^*} g(y)dy = \lim_{\Delta y \to 0} \sum_{-\infty}^{y_t^*} g(y)\Delta y = \\ \sum_{-\infty}^{y_t^*} \lim_{\Delta y \to 0} G(y + \Delta y) - G(y) = \\ \sum_{-\infty}^{y_t^*} \lim_{\Delta y \to 0} \sum_{n=1}^{\infty} \frac{G^{(n)}(y)}{n!} * (\Delta y)^{(n)} .$$

Where $g(.)$ stands for the first derivative of $G(.)$ with respect to its argument, and $G^{(n)}(.)$ for the $n - th$ derivative. It is worth noting that $G(y)$ disappears, and only its derivatives are left.

If we want the contribution of each of the terms to be additive and to sum up to the total output ([10] refer to an heatmap with this property as a conservative one), then we need to truncate the Taylor expansion to the first order.

The numerical integration is required to avoid loosing accuracy when truncating the Taylor expansion.

Clearly, the higher is the degree of non linearity of the function $G(.)$, the bigger will be the gain in accuracy provided by the numerical integration.

Hence,

$$G(y_t^*) \approx \sum_{-\infty}^{y_t^*=\beta * x_t} g(\beta' x) * (\beta' \Delta x) = \\ \sum_{-\infty}^{y_t^*=\beta * x_t} g(\beta' x) * (\beta_1 \Delta x_1 + ... + \beta_K \Delta x_K).$$

For a small enough $\Delta x_k$ the approximation error gets negligible.

It is left to define the starting point of the numerical integration, i.e.: the initial root point for the Taylor expansion.

Assuming $G : X \to [a, b]$, for observation $x$ the initial root point is found by solving the following constrained minimization problem [10, 16]:

$$min_\theta \parallel \theta - x \parallel^2 \text{ subject to } G(\theta) = a \text{ and } \theta \in X \tag{2}$$

This minimization problem is not necessary solvable due to its possible non-convexity.

[4] As shown for logit models in [5].

[5] Classification versus regression.

Nevertheless, when $G(\theta)$ is monotonic in its inputs and $| a |< \infty$ the problem is trivial and a candidate for the initial root point is the minimum of the set $X$ collecting all observations[6].

Thanks to the numerical integration, any possible loss in accuracy due to choosing a starting point too distant from the actual observation vector is reduced to a very small and negligible number.

In case of singe layer classifier $G : X \rightarrow [0, 1]$, the algorithm implements as follows.

**Algorithm 1.** *Steps to explain simple classifiers*

1. *Store the parameters from your model estimated on standardized X*
2. *If there is a negative coefficient, map it to positive by multiplying both the coefficient and the variable by* $-1$
3. *(Searching Initial Root Point) Calculate the value of* $y^*$ *at the set of points* $X_{min} = -c = min(X)$ [7]*, this is your lower bound for the sum, while the upper bound* $y^*_t = \beta * x_t$ *is provided by the observations on which you want to apply the decomposition. Check that* $y_{min} = G(\beta * X_{min}) = 0$ *if not increase c until this condition is met*
4. *Define the number of grid points (you can make it proportional to the distance between the current X and* $X_{min}$*)*
5. *Generate the matrix of observations such that for every* $x_i$ *you have all the values from the smallest (*$X_{min}$*) to the current observation* $x_{t,i}$ *with constant delta dependent on the number of grid points defined above and on the distance between the root point and the current observation*
6. *Apply formula in equation (1). Sum up all the contributions and check the difference with the prediction of your model. If it is too high, then increase the number of grid points*

### 2.2. *Deep neural networks*

In this section we will apply the method to explain a prediction coming from a feed-forward deep neural network (DNN) with respect to its input data. To note that the algorithm proposed does not apply only to classification problems. Indeed, in the next section we will present an application to a regression problem.

While we provide examples for simple feed-forward deep neural networks, with relatively minimal adjustments to the algorithm, applications to more complex neural nets (e.g.: convolutional networks and recurrent neural networks) can be carried out.

In principle, any non linear model that can be decomposed into trivial functions (what this stands for will be clearer soon), and non-trivial functions satisfying assumptions made in Claim 1 can have its predictions decomposed as discussed in this article.

When the DNN under analysis satisfies the conditions to deliver a unique solution to the optimization problem in (2)[8], or a theoretical grounded baseline for the right hand side variables is provided by the researcher, then one can apply the IG method of [15].

Otherwise, the algorithm for the general case is presented here below.

**Algorithm 2.** *Steps to explain deep neural networks: deep explanator (DE)*

*Let's first define two sets of link functions. Being* $\Theta$ *the set of all the link functions and* $\Omega \subset \Theta$ *the one with trivial functions (i.e.: ones for which we do not need to use Taylor*[9]*), we can define the set of non-trivial link functions:* $\Phi = \Theta \setminus \Omega$*.*

1. *Store the parameters from your model estimated on standardized X*
2. *Starting from the first layer, loop over its neurons and compute the contribution of each input to the current output*
3. *When computing the contribution check if the function is in set* $\Phi$*; apply the stepwise Taylor expansion only in this case (Algorithm 1)*
4. *Do so for every layer*
5. *You will get* $NH + 1$ *matrices for each observation; where NH is the number of hidden layers*[10].
6. *Finally, multiply all these matrices and you will get the contribution of each input to the output*
7. *Sum over the contributions and check whether the approximation error is too high. If so, then increase the number of grid points*

Additionally, and in case of a classification problem, one could have estimated an optimal threshold $a$ according to some method (as in [14] or [1]), and therefore be

[6] In the rare situation where $G(\theta) > a$ with $\theta_i = min(X)\ \forall i = 1, ..., K$, one can decrease $\theta_i\ \forall i$ until $G(.)$ is equal to its lower bound. On the other side, when $x_0$ is too low, no bias is added. Indeed, given $x_0 \in \Re^K : G(x_0) = a$ then $g(x) = 0\ \forall x < x_0$.

[7] Where $X$ is the matrix of right hand side standardized observations.

[8] One method to achieve this is by imposing constraints in the network structure as in [6].

[9] e.g.: linear or ReLU and its generalizations; more in general, any link function whose derivatives are constant once the output of that link function is known.

[10] All matrices but the one of the first layer need to be rescaled in order to isolate the neuron impact from its size.

interested in explaining the portion of $y \in [a, 1]$. This can be achieved by simply finding the root point $c$ for the neurons in the last hidden layer that makes the link function of the output layer $G(.)$ computed at this point equal to the desired threshold $a$. Then, back propagation of this process in all the neural network will provide the root point for the variables of interest at the input layer.

## 3. Empirical applications

In this section we will apply Algorithm 2 to two problems: regression and classification. The purpose of this section is to show how an intuitive graphical representations can be carried out.

### *3.1. Application to a regression problem*

For the regression problem we consider the American Housing Survey dataset. This is the same dataset used in [12], and consists of 162 features with 51808 observations.

We fit a DNN with 2 hidden layers. Namely, a ReLU type link function connects the initial 162 inputs to the first hidden layer with 16 neurons, then those neurons are connected via a sigmoid link to the second hidden layer with 4 neurons, which are finally mapped to the total output via a linear aggregation.

Figure 1 shows a stacked bar chart for the decomposition of the aggregate output, together with predicted and actual standardized values for 20 observations[11]. The 162 initial features are mapped to 6 groups according to the topic; while, the contribution of biases is cumulated.

### *3.2. Application to a classification problem*

When it comes to classification, we use the Pima Indians Diabetes dataset from the National Institute of Diabetes and Digestive Kidney Diseases[12], as in [9][13].

In this application we decompose the entire prediction $y \in [0, 1]$.

In the dataset there are a total of 8 features. We use again a DNN with 2 hidden layers. A ReLU function maps those 8 features to 6 neurons, then sigmoid links reduce them to 4, ultimately another sigmoid makes the final aggregation.

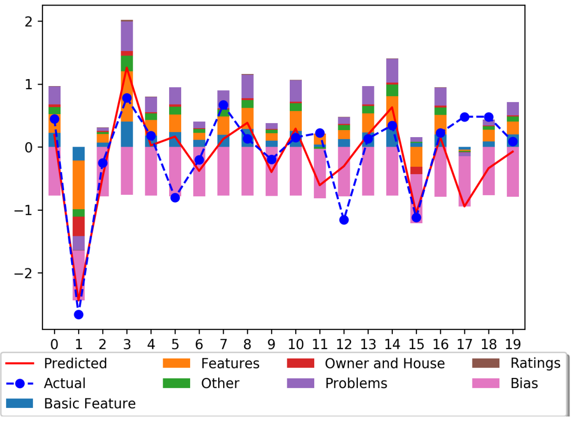

Figure 1: Explanation of DNN predictions - regression using the American Housing Survey dataset

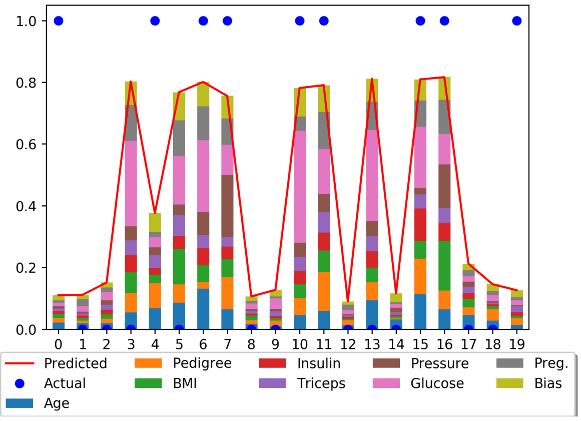

Figure 2: Explanation of DNN predictions - classification using the Pima Indians Diabetes dataset

Figure 2 presents the decomposition on the first 20 observations of the test set (20% of the overall 768 data points), here as well we cumulate the contribution of the biases.

## 4. Conclusion

In this article, after briefly discussing current methods used to explain predictions in some non linear models, we have proposed a new approach that merges two of the ones present in the literature. In particular, the deep explanator (DE) presented in Algorithm 2 has been motivated by the weaknesses found in IG and DTD.

Namely, compared to the IG, it facilitates the search of

[11] Those 20 are taken as the first observations of the test set part, which accounts for 20% of the overall data.

[12] The dataset can be found at this GitHub link: https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv. Original donor is Vincent Sigillito.

[13] We cite only the most recent paper and refer to the bibliography in there for the interested reader.

the baseline (initial root point) by exploiting the monotonicity of each of the non-trivial link functions. More so, it maintains high accuracy, something not always shared by the plain DTD.

Nevertheless, and as pointed out also by [15] among others, while there are advances on explaining a prediction made by non linear models such as deep networks, few progresses have been made in the general understanding of the interactions captured by those models.

## References

[1] Alessi, L., Detken, C., 2011. Quasi real time early warning indicators for costly asset price boom/bust cycles: A role for global liquidity. European Journal of Political Economy 27, 520–533.

[2] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W., 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one 10, e0130140.

[3] Bazen, S., Joutard, X., 2013. The taylor decomposition: A unified generalization of the oaxaca method to nonlinear models. Technical Report , 2013–32.

[4] Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N., 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , 1721–1730.

[5] Denk, C.E., Finkel, S.E., 1992. The aggregate impact of explanatory variables in logit and linear probability models. American Journal of Political Science 36, 785–804.

[6] Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R., 2009. Incorporating functional knowledge in neural networks. Journal of Machine Learning Research 10, 1239–1262.

[7] Goodman, B., Flaxman, S., 2017. European union regulations on algorithmic decision-making and a right to explanation. AI Magazine 38, 50–57.

[8] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D., 2018. A survey of methods for explaining black box models. ACM computing surveys (CSUR) 51, 93.

[9] Maniruzzaman, M., Rahman, M.J., Al-MehediHasan, M., Suri, H.S., Abedin, M.M., El-Baz, A., Suri, J.S., 2018. Accurate diabetes risk stratification using machine learning: role of missing value and outliers. Journal of medical systems 42, 92.

[10] Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R., 2017. Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition 65, 211–222.

[11] Montavon, G., Samek, W., Müller, K.R., 2018. Methods for interpreting and understanding deep neural networks. Digital Signal Processing 73, 1–15.

[12] Mullainathan, S., Spiess, J., 2017. Machine learning: an applied econometric approach. Journal of Economic Perspectives 31, 87–106.

[13] Ribeiro, M.T., Singh, S., Guestrin, C., 2016. Why should i trust you?: Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining , 1135–1144.

[14] Sarlin, P., von Schweinitz, G., 2017. Optimizing policymakersloss functions in crisis prediction: Before, within or after? Macroeconomic Dynamics , 1–24.

[15] Sundararajan, M., Taly, A., Yan, Q., 2017. Axiomatic attribution for deep networks. Proceedings of the 34th International Conference on Machine Learning-Volume 70 , 3319–3328.

[16] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 .

[17] Tsang, M., Cheng, D., Liu, Y., 2017. Detecting statistical interactions from neural network weights. arXiv preprint arXiv:1705.04977 .