Adaptive splines-based logistic regression with a ReLU neural network

Marie Guyomard, Susana Barbosa, Lionel Fillatre

Adaptive splines-based logistic regression with a ReLU neural network

Marie GUYOMARD[1], Susana BARBOSA[2] and Lionel FILLATRE[1]
[1] University Côte d'Azur, Laboratory I3S, France
[2] University Côte d'Azur, Laboratory IPMC, France

Corresponding author: guyomard@i3s.unice.fr

Abstract *This paper proposes a non-linear binary classification model. Although linear classification methods are very popular in the field of personalized medicine because of their interpretability, they have proven to be too restrictive. Doctors are convinced of the need to quantify threshold effects for better predictions. Nevertheless, non-linear methods that can be found in the state of the art are not able to automate the segmentation of variables (Segmented Logistic Regression) or are difficult to interpret (Random Forests or Neural Networks). We propose a Neural Network that fully realizes a non-linear logistic regression. The score function of the logistic regression, initially linear, is replaced by a piecewise linear function, modeled by spline functions. The particular architecture of this network automates the segmentation of the variables and guarantees its operational relevance as well as the explicability of its calculated predictions.*

Keywords Non-linear Classification, Splines approximation, Neural Networks.

1 Introduction

The use of Artificial Intelligence in the medical field continues to progress. Machine learning models for classification allow in many practical cases to avoid invasive methods, such as biopsies, to provide an accurate diagnosis. The use of Logistic Regression (LR) is widespread. For example, in [1] the LR is used to predict the development of non-alcoholic cirrhosis. Unlike methods such as boosting, random forests, and neural networks (NNs), the model estimated by LR is easily interpretable since it depends on a linear combination of explanatory variables.

Nevertheless, doctors believe that incorporating nonlinear phenomena in the modeling, such as threshold effects on certain variables, would increase the predictive performance. For example, a significantly high cholesterol level could be a risk factor for developing a disease while a low cholesterol level would decrease this risk. For this purpose, a promising track is the use of a LR that exploits a non-linear model based on piece-wise linear splines. These splines divide the domain of definition of the explanatory variables into several segments and, on each segment, perform a linear approximation.

The major difficulty of using an approximation with splines relies on the choice of the bounds of each segment, called knots of the spline. It has been shown in [2,3] that jointly optimizing the knots and the linear approximation associated with each segment is difficult. The fixed splines based-LR avoids these optimization issues by fixing a priori the number and the value of the knots. The choice of knots is therefore rather arbitrary and the model becomes frozen. An alternative is the MARS (Multivariate Adaptative Regression Splines) model which proposes an adaptive method to compute the knots. Those are computed recursively in order to progressively improve the performance of the model. The global performance of the method is not directly optimized but in a greedy and sub-optimal way. Recently, a third approach has emerged : it establishes a rigorous bridge between deep neural networks (NNs) and the theory of piecewise linear spline functions. Contrary to the MARS model, in NNs a global criterion is minimized to learn the segmentation. Unfortunately, the segmentation produced by an NN is very complex and therefore it becomes impossible to easily interpret the impact of explanatory variables in the prediction.

This work has been supported by the French government, through the UCA DS4H Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-17-EURE-0004.

The main contribution of this paper consists in developing an NN inspired by the MARS model in order to combine the advantages of MARS and NNs: the minimization of a global criterion to obtain an NN with an easily explicable architecture. A second contribution consists in proposing an explicit algorithm to train this NN. Indeed, the authors in [4] demonstrate that it is possible to approximate MARS models by NNs but their approach, totally theoretical, does not propose any algorithm to train the NN. Finally, we compare our NN to the previously mentioned methods on a simulated and a real datasets. Our NN presents comparable or superior performances in prediction to the other approaches while being fully explainable.

This paper is structured as follows. The section 2 introduces the studied prediction problem. The section 3 describes our NN architecture. The section 4 presents the experimental results. Finally, the section 5 concludes the article.

2 Problem Statement

We have N independent and identically distributed pairs $(x^{(i)}, y^{(i)})$ where $x^{(i)}$ is the vector of explanatory variables and $y^{(i)} = \{0, 1\}$ is the binary label to predict. The notation (X, Y) denotes the pair of random variables from which the pairs $(x^{(i)}, y^{(i)})$ are derived. These data are used to train and test all the models implemented in the article.

2.1 Bayesian binary classification

A Bayesian Maximum a Posteriori (MAP) classifier assigns a label y to a sample $x = [x_1, \ldots, x_d]$ based on the decision rule $\delta : \mathbb{R}^d \mapsto [0, 1]$ defined by $\delta(x) = \hat{\mathbb{P}}(Y = y|X = x)$ where $\hat{\mathbb{P}}(Y|X)$ is an estimate of the conditional a posteriori probability. The LR is the most widely used decision rule for this kind of problem in the medical field. It is written as

$$\delta^{\mathrm{RL}}(x) = \sigma(\psi(x)) = \frac{1}{1 + \exp\left(-\psi(x)\right)}, \tag{1}$$

where $\sigma(\cdot)$ is the logistic function and $\psi(x)$, called the score function, is a linear function $\psi(x) = \beta^\top x$ where $\beta = [\beta_1, \ldots, \beta_d] \in \mathbb{R}^d$ is a vector of coefficients and $\beta^\top$ denotes the transposed vector β. Each coefficient β_i quantifies the impact of the i^{th} component of the vector x on the probability of choosing the class $y = 1$. This model is very appreciated for its simplicity and its explicability.

2.2 MARS score function of order 1

In order to obtain a non-linear but still explainable score function, a relevant modeling is brought by the MARS approach [5]. This model is based on an approximation with adaptive splines of the score function:

$$\psi^{\mathrm{MARS}}(x) = \sum_{m=1}^{M} \beta_m h_m(x), \tag{2}$$

where $h_m(x)$ is a spline function of the form

$$h_m(x) = \left[s_m(x_{v(m)} - b_m)\right]_+ \tag{3}$$

$$= \begin{cases} \max\{0, x_{v(m)} - b_m\}, & \text{if } s_m = 1, \\ \max\{0, b_m - x_{v(m)}\}, & \text{if } s_m = -1. \end{cases} \tag{4}$$

The notation $[t]_+ = \max\{0, t\}$ denotes the ReLU function. The function $h_m(x)$ depends on the $v(m)^{\text{th}}$ component $x_{v(m)}$ of the vector x. The real b_m is the knot of the spline. The integer $s_m \in \{-1, 1\}$ used in conjunction with the ReLU function allows to cancel the left part or the right part of $h_m(x)$ as explained in (4).

The MARS approach learns the $h_m(x)$ functions sequentially. At each iteration $m \in \{1, \ldots, M\}$, the spline function $h_m(\cdot)$ that best reduces the learning error is added. The recursive and adaptive segmentation of the MARS approach is thus similar to that of decision trees. If $v(m) \neq k$ for all m, then the k^{th} component of x will never be included in the model. The MARS approach is based on a

greedy optimization algorithm whose global optimality is not established. The recursivity of the model makes the segmentation of variables uncontrollable. It is possible that a same variable is segmented a large number of times. However, we know from doctors' feedback that over-segmenting a biological variable is not relevant.

2.3 ReLU Neural Networks

The non-linear approach that is currently very widespread is based on a score function produced by a ReLU NN [6]. In this paper, we consider only single-hidden-layer NNs that are written as

$$\psi^{\text{NN}}(x) = \beta_0 + \beta^\top [Wx + b]_+, \tag{5}$$

where $\beta \in \mathbb{R}^p$, $W \in \mathbb{R}^{p \times d}$ is the matrix of weights, $b \in \mathbb{R}^p$ is the vector of biases, and $[z]_+$ denotes the vector z where the function $[\cdot]_+$ has been applied to each component. The hidden layer has p neurons. According to [7], deep networks with the ReLU activation function introduce a partitioning of $\mathbb{R}^d$ that is equivalent to an approximation with multidimensional splines. However, this partitioning is very complex and generally unexplainable. The figure 2-e illustrates this partitioning when $d = 2$ with $p = 30$ neurons. The straight lines, almost always oblique, intersect and cut the space into polyhedra with very diverse geometrical shapes. This division explains the flexibility of an NN but also why an NN is considered as a "black box".

3 Neural Network NN-MARS

In order to benefit from the advantage of training a neural network (minimization of a global criterion with a gradient descent) and to keep an explicability close to the MARS model, we propose in this section a piecewise continuous score function modeled with a NN. Our model $\psi^{\text{NN-MARS}}(x)$ is written as

$$\psi^{\text{NN-MARS}}(x) = \beta_0 + \sum_{j=1}^{d} g_j(x_j), \tag{6}$$

$$g_j(t) = \beta_{j1}[b_{j1} - t]_+ + \beta_{j2}[t - b_{j2}]_+, \quad t \in \mathbb{R}. \tag{7}$$

In (6), the real nonlinear function $g_j(\cdot)$ is applied to x_j, the j^{th} component of the vector $x \in \mathbb{R}^d$. The function $g_j(t)$ corresponds to a pair of neurons working together: the first neuron is a non-zero spline before the knot value b_{j1} and the second one is a non-zero spline after the knot value b_{j2}. As a result, $g_j(t)$ models functions with a pattern composed of 3 linear segments as illustrated in the gray box on Figure 1. In this figure, the variable X_1 can represent for example the weight of a patient. Being underweight ($X_1 < b_{11}$) or overweight ($X_1 > b_{12}$) increases the probability of developing the pathology. In contrast, between these two intervals, the impact of weight on disease is negligible. The segmentation of the network is therefore controlled: at most 3 segments are created for each descriptive variable.

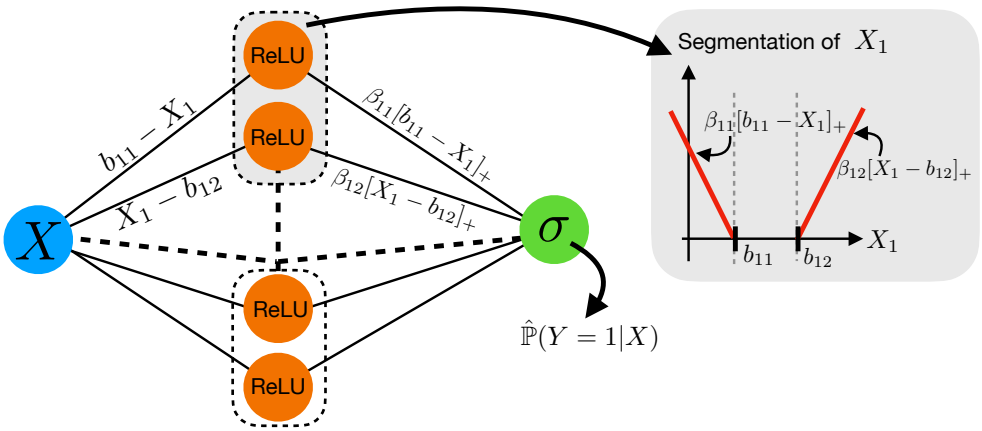

Fig. 1. Architecture of NN-MARS: the inputs in blue, the hidden layer in orange and the estimated labels in green.

Furthermore, the nature of the segmentation operated by the NN-MARS is also controlled. Contrary to classical ReLU NNs which create oblique regions by linearly combining the components of x, the NN-MARS cuts the components of x independently. It relies on hyperplanes that are orthogonal to the canonical basis of the $\mathbb{R}^d$ space, just as decision trees or the MARS model do, as illustrated in Figure 2-f. The partitioning of the $\mathbb{R}^d$ space is done with hypercubes and not polyhedra with complex shapes. The decision rule obtained is easily interpretable: the score function is linear on each hypercube. In practice, this is equivalent to performing a local LR by thresholding the components of the vector x. The function $\psi^{\text{NN-MARS}}(x)$ models the impact of each component x_i with a specific non-linear profile as illustrated in Figure 4. Overall, the NN-MARS is composed of $2d$ hidden neurons. Figure 1 shows that the hidden neurons operate in pairs. The training of NN-MARS is done with an ordinary gradient descent using the cross-entropy as loss function.

4 Experiments

We compare the performance and explainability of NN-MARS to decision trees (DTs), LR Natural Cubic Splines (LR NCS) [8, section 5.2] with fixed knots using uniform quantiles, MARS, and classical ReLU NNs. A 5-folds cross validation is realised for each tested method, such that the training sample is composed of 70% of the data and thus the validation set of the remaining 30%. All the experiments are implemented in Python with *scikit-learn* functionalities and *Pytorch* ones for the NNs. The computation times displayed in the article are reasonable since a computation server with a GPU is used.

Simulated data: We simulated $x \in \mathbb{R}^2$ data in order to visualize the estimated decision boundaries. We seek to predict the probability of developing a pathology as a function of cholesterol level (x_1) and weight (x_2). When the cholesterol level is low, the patient is more protected against the disease. On the other hand, when it is high, the patient is more at risk. Finally, being underweight or overweight increases the probability of developing the pathology. Since in real applications the boundary is noisy, we define the labels using a Bernouilli distribution with the estimated probability of developing the disease as parameter. The dataset is composed of 2000 patients, such that 1000 are sick (class 1).

Figure 2 presents the six methods compared. The red border represents the decision boundary of the tested method. The black lines represent the edges of the partitioning produced by the tested method. The deep NN (Fig. 2 - NN ($p = 30$)) is the best performing model because its estimated boundary is the closest to the simulated one. Nevertheless, the interpretation of its decision rule is too complex because the $\mathbb{R}^2$ space is partitioned into many polyhedra, some of which are not very useful for approximating the boundary. The DT, the LR NCS, MARS as well as NN-MARS segment the variables by hyperplanes making easier the interpretation of the decision rule (Fig. 2). Indeed, the partitioning depends on only one variable, contrary to the traditional NNs (Fig.2 - NN ($p = 4$) & NN($p = 30$)) that lead to oblique regions. For instance, with the proposed method, we can explain to the doctors that we have to take into account a 3 parts-segmentation of the cholesterol level : one for the patients with a level lower than -1.6, one for those with a value bigger than -0.3 and finally the intermediate ones. The MARS model with $M = 10$ (Fig.2 - MARS) defines 5 knots for the variable x_1 and 5 for x_2 in very close values. NN-MARS achieves the same performance as LR NCS with fewer spline functions, which justifies the need in automating the segmentation and not fixing the knots a priori.

The NN-MARS performs almost as well as the ReLU NNs while requiring fewer parameters to be estimated and ensuring a good explainability. Indeed, 9 parameters have to be optimized by the NN-MARS whereas the ReLU NNs with $p = 4$ and $p = 30$ require to estimate respectively 17 and 121 parameters. Although the NN-MARS estimates fewer parameters, the computation time required to converge to an optimal modeling is higher for the proposed method : 178($\pm$1) seconds for one fold are required whereas the ReLU NNs with $p = 4$ and $p = 30$ take respectively 85($\pm$26) and 120($\pm$27) seconds. The standard deviations of computation time are higher for the traditional NNs because of the early stopping process that occur between 4000 and 8000 iterations at each fold, while the NN-MARS does not necessitate it and runs over 10000 epochs.

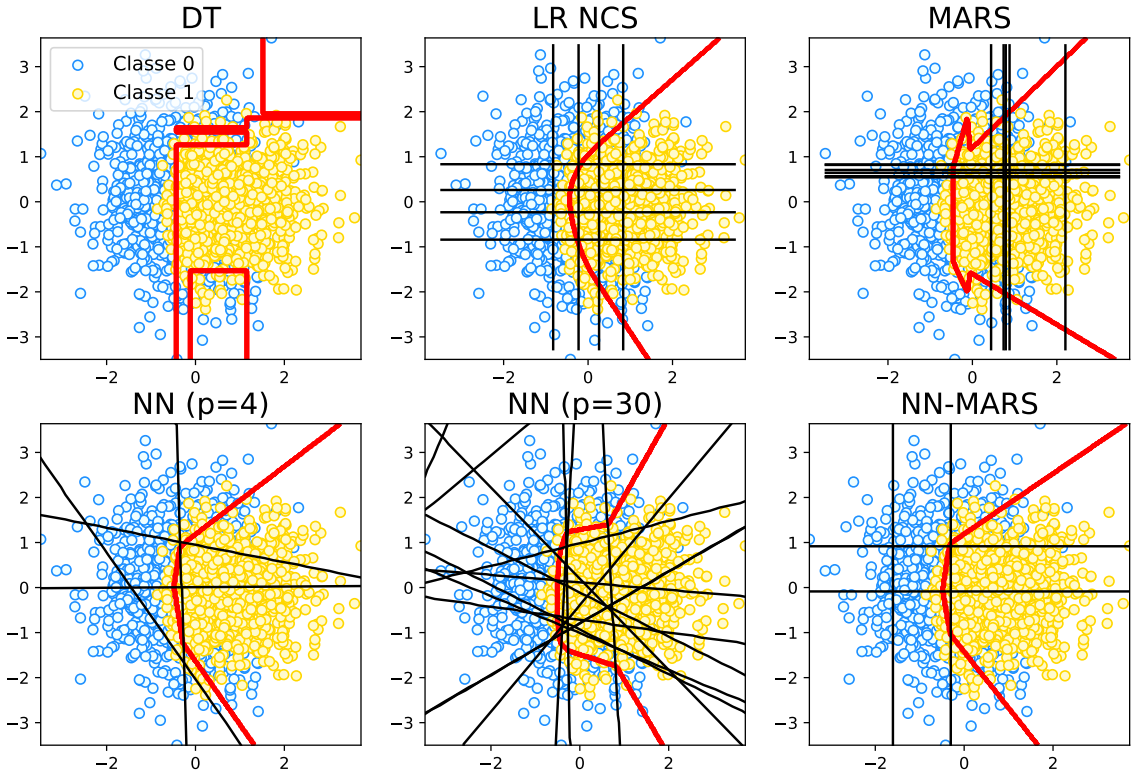

Fig. 2. Results on simulated data for DT, LR NCS, MARS model with $M = 10$, NN with $p = 4$ in (5), NN with $p = 30$ in (5), NN-MARS (4 nodes, since $d = 2$). Caption: estimated boundary in red, segmentation in black.

Finally, an experiment to test the scalability of the proposed method is presented on the figure 3. The AUC (Area Under the Curve) is a criterion appreciated by the doctors [9]. Where accuracy provides information about the predictive power of a model, the AUC gives an idea of the Sensibility and the Specificity of a predictive model, that is to say the capability of a tool for identifying a binary signal (sick or not in our case). The obtained average AUC over the 5-folds cross validation is displayed on both training and test samples for the NN-MARS, the ReLU NN with $p = 4$ and the ReLU NN with $p = 30$ according to different samplings. The proposed method (Fig. 3 curve in blue) obtains an AUC equivalent or even higher to the traditional ReLU NNs on the testing sample for all the number of patients tested. Indeed, traditional ReLU NNs tend to over-fit when only few patients are available in the training sample. The freezing of some parameters in the neural architecture of NN-MARS does not affect its predictive performance while improving the interpretability of its decision rule, even when a few number of patients is available in a cohort.

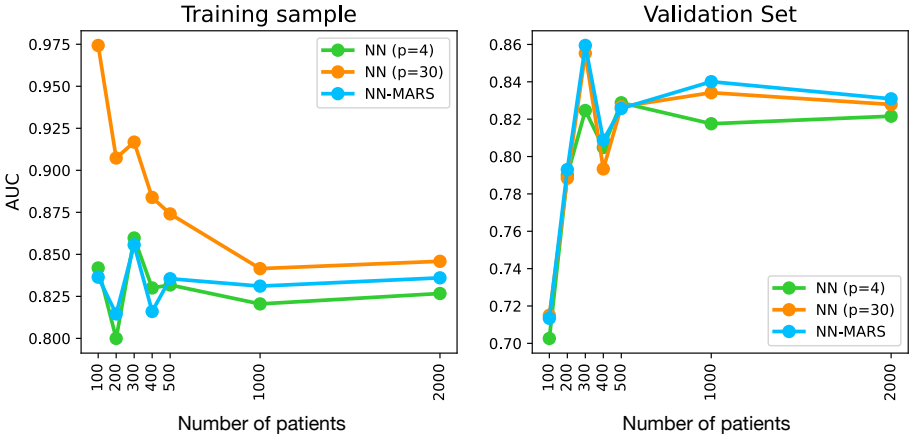

Fig. 3. AUC score (y-axis) on simulated data with different number of patients (x-axis) for the NN-MARS in blue, ReLU NN with $p = 4$ in green and ReLU NN with $p = 30$ in orange on training and validation samples.

Real data: We compared the performance of NN-MARS to other methods on a real data set. The main objective of the "Parkinson" database [10] is to detect people with Parkinson's disease from voice recordings of $N = 195$ patients ($24,6\%$ with Parkinson). We kept $d = 16$ biomedical measures of voice, such as maximum, average, and minimum voice frequencies. The NN-MARS is composed of 32 neurons. The training of the NNs is stopped when the error on the test data does not decrease anymore in order to avoid overfitting. The results are detailed in the table 1.

	Training set		Test set	
	Accuracy	AUC	Accuracy	AUC
LR	0.85 (0.02)	0.87 (0.02)	0.76 (0.01)	0.80 (0.06)
DT	0.91 (0.02)	0.94 (0.02)	0.88 (0.01)	0.77 (0.03)
LR NCS	0.90 (0.02)	0.94 (0.01)	0.82 (0.03)	0.87 (0.05)
MARS	0.90 (0.03)	0.91 (0.06)	0.82 (0.04)	0.89 (0.04)
NN ($p = 16$)	0.87 (0.04)	0.91 (0.07)	0.81 (0.06)	0.88 (0.07)
NN ($p = 70$)	0.86 (0.04)	0.91 (0.07)	0.83 (0.06)	0.88 (0.07)
NN-MARS	0.87 (0.01)	0.92 (0.03)	0.83 (0.05)	0.91 (0.05)

Tab. 1. Results of predictive performance on real data (mean and standard deviation in parentheses): the DT, the LR NCS, the 16-neurons NN, the 70-neurons NN, and the NN-MARS (6).

The LR is less efficient (76% accuracy on the test sample) than the other methods, which highlights the importance of introducing non-linearity in the modeling. Among all the non-linear methods tested, NN-MARS obtained the best AUC on the test sample (91%). The automation of the segmentation in MARS and NN models explains their higher AUCs than the LR NCS. The standard deviations show that the NN-MARS is more stable than the conventional NNs.

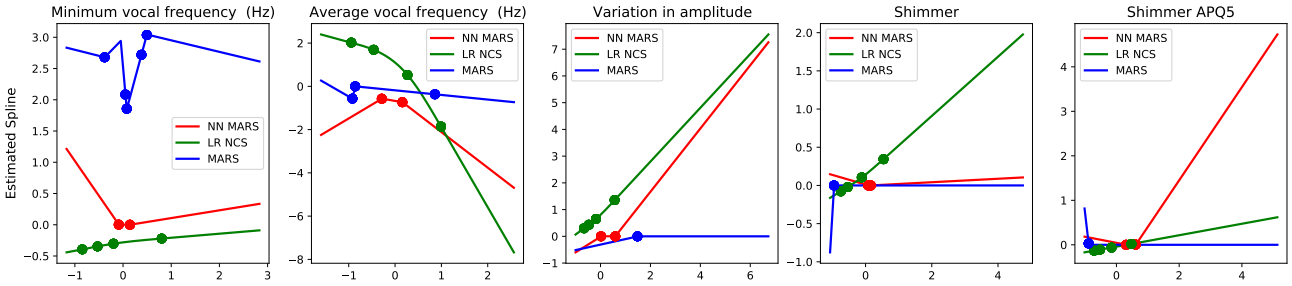

Fig. 4. Estimated splines on Parkinson dataset for the following features: Minimum voice frequency, Average voice frequency, Amplitude variation, Shimmer, Shimmer APQ5. Caption: NN-MARS in red, MARS in blue, LR NCS in green.

An advantage of the NN-MARS is the graphical interpretability of the results. Contrary to traditional ReLU NNs, the proposed method performs a partitioning with hyperplanes and thus the univariate estimated splines can be viewed. Figure 4 shows the estimated splines for 5 predictive variables. Each spline models the impact of the corresponding variable on the classification score. The estimated knots are represented by the points on the curves. The different methods find similar profiles for the variables but still quite dissimilar. For example, the NN-MARS (curves in red in Fig. 4) estimates that a low average vocal frequency increases the risks of developing Parkinson, since its estimated spline increases as the value of this feature decreases. From a certain level, the augmentation of the vocal frequency decreases the risk of being classified as sick. Moreover, this figure highlights the limits of the greedy learning of the MARS method. Only 5 variables are segmented, one of them 5 times (Fig. 4-Minimum vocal frequency). The MARS model is not able to increase its predictive performance by adding a new spline function, while it is possible to find a better performing classification rule, as demonstrated by the NNs in Table 1.

5 Conclusion

This paper develops an explainable neural network for nonlinear binary classification. Threshold effects are included in the modeling of the problem by approximating splines in the hidden layer of the network. The particular architecture of this network allows to control the segmentation of the variables and also to produce an easily understandable decision rule. Thus, this model is adapted

to medical problems and to the expectations of specialists in this field. In future works, it would be interesting to include categorical variables and interactions between variables in order to improve predictive performance.

References

[1] Marie Guyomard, Dann J. Ouizeman, Renaud Schiappa, Cyprien Gilet, Jocelyn Gal, Emmanuel Chamorey, Stéphanie Patouraux, Thierry Piche, Albert Tran, Philippe Gual, Antonio Iannelli, Lionel Fillatre, and Rodolphe Anty. Diagnostic non-invasif de la nash fibrosante à l'aide de l'intelligence artificielle. *AFEF (Société Française d'Hépatologie)*, 2020.

[2] Douglas M Hawkins. On the choice of segments in piecewise approximation. *IMA Journal of Applied Mathematics*, 9(2):250–256, 1972.

[3] Asher Tishler and Israel Zang. A new maximum likelihood algorithm for piecewise regression. *Journal of the American Statistical Association*, 76(376):980–987, 1981.

[4] Konstantin Eckle et al. A comparison of deep networks with relu activation function and linear spline-type methods. *Neural Networks*, 110:232–242, 2019.

[5] Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67, 1991.

[6] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.

[7] Randall Balestriero et al. A spline theory of deep learning. In *International Conference on Machine Learning*, pages 374–383. PMLR, 2018.

[8] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[9] Xiao-Hua Zhou, Donna K McClish, and Nancy A Obuchowski. *Statistical methods in diagnostic medicine*. John Wiley & Sons, 2009.

[10] Max Little et al. Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *Nature Precedings*, 2008.