Feature-Aware Drop Layer (FADL): A Nonparametric Neural Network Layer for Feature Selection

Manuel Jesús Jiménez-Navarro[1](✉), María Martínez-Ballesteros[1],
Isabel Sofia Sousa Brito[2], Francisco Martínez-Álvarez[3],
and Gualberto Asencio-Cortés[3]

[1] Department of Computer Science, University of Seville, 41012 Seville, Spain
{mjimenez3,mariamartinez}@us.es

[2] Department of Engineering, Polytechnic Institute of Beja, Beja, Portugal
isabel.sofia@ipbeja.pt

[3] Data Science and Big Data Lab, Pablo de Olavide University, 41013 Seville, Spain
{fmaralv,guaasecor}@upo.es

Abstract. Neural networks have proven to be a good alternative in application fields such as healthcare, time-series forecasting and artificial vision, among others, for tasks like regression or classification. Their potential has been particularly remarkable in unstructured data, but recently developed architectures or their ensemble with other classical methods have produced competitive results in structured data. Feature selection has several beneficial properties: improve efficacy, performance, problem understanding and data recollection time. However, as new data sources become available and new features are generated using feature engineering techniques, more computational resources are required for feature selection methods. Feature selection takes an exorbitant amount of time in datasets with numerous features, making it impossible to use or achieving suboptimal selections that do not reflect the underlying behavior of the problem. We propose a nonparametric neural network layer which provides all the benefits of feature selection while requiring few changes to the architecture. Our method adds a novel layer at the beginning of the neural network, which removes the influence of features during training, adding inherent interpretability to the model without extra parameterization. In contrast to other feature selection methods, we propose an efficient and model-aware method to select the features with no need to train the model several times. We compared our method with a variety of popular feature selection strategies and datasets, showing remarkable results

Keywords: Feature selection · Neural network · Classification · Regression

https://doi.org/10.1007/978-3-031-18050-7_54

1 Introduction

Neural networks have shown to perform well in supervised learning tasks in a variety of domains such as healthcare, time-series forecasting, artificial vision, etc. Their capability in unstructured data, such as images or text, has been demonstrated in the literature and real applications. However, structured data modeling with neural networks is evolving rapidly and achieving competitive results [1].

Although neural networks produce competitive results, some well-known issues remain: curse of dimensionality, performance limitations, noise data or lack of interpretability are some examples. Feature selection is an effective technique that reduces the impact of these issues by selecting a subset of features [10].

Because feature selection reduces the number of features, the resources consumed by the neural network are considerably reduced as it requires less memory, less time to fit and less inference time. This is especially useful nowadays, with the increasing number of data sources and the application of feature engineering.

Feature selection methods can be divided into two groups: model-free methods and model-aware methods. Model-free methods employ data analysis to determine the importance of features. This type of method is very efficient, but does not consider the model in the selection. Model-aware methods analyze the importance of the features learned by the model. However, these methods can involve more processing time, but the selected features can be related to the model.

Metaheuristics-based is a model-aware selection method which must fit a model several times consuming a prohibitive amount of resources. In most of the cases, the selection is limited, resulting in a suboptimal feature selection, which might impair the performance, efficacy and interpretability of the model.

Most feature selection approaches incorporate some or several hyperparameters such as a threshold to select the best ranked features or a predefined number of iterations in the search space. Such types of methods are called parametric. Poor hyperparameter selection can turn out to be suboptimal selection, whereas optimal hyperparameter selection implies repeating the selection method using more computation time.

In this paper, we propose a feature selection method for neural networks optimized by backpropagation. We consider a scenario with limited resources in which we can sacrifice some efficacy to reduce the number of features. The feature selection method is implemented in the neural network as an additional layer, hereinafter called FADL, after the input layer. The proposed method is a non-parametric model-aware selection method that adds an inherent interpretability to the model without extra parameterization. This interpretability is understood by knowing which features the model uses once it is trained.

We compared our method with three different model-free feature selection methods called: Linear, Correlation and Mutual information [4]. All the selected model-free methods establish a score to the features which is used to determine which ones are relevant. We use these methods in the comparison because they are usually faster than model-aware methods as it must optimize just the threshold to determine which features are relevant. In addition, model-free methods scale well with when the number of features increase. On the contrary, model-aware methods may need to explore the feature space to obtain the best selection. Consequently, when the feature space increases, the resources needed increase exponentially.

In contrast to other feature selection methods, our method obtains competitive results more efficiently since it does not need to train the model several times. Although, our method is scalable as the feature space exploration is made during the training time. These methods are compared in five classification datasets and four regression datasets, obtaining a considerable improvement in the effectiveness of regression datasets.

The remainder of this paper is organized as follows. Section 2 describes the related work and the key differences from our work. In Sect. 3, the methodology and each of its components are explained. In Sect. 4, the experimentation is explained including the experimental settings, results and discussion. Finally, in Sect. 5, we indicate some conclusions from the method and future work.

2 Related Works

Feature selection is a well-known technique in the literature, and there exist numerous methods that can be applied to almost any neural network structure. In this paper, we will focus on feature selection methods developed for neural networks. There are two types of methods used to select features in neural networks: meta-models and custom regularization. Meta-model methods are model-aware feature selection while custom regularization methods are model-free method.

In Verikas et al. [15], Khemphila et al. [7] and [11] a meta-model is used to select the features. This method uses a workflow that finds the optimal feature selection for the model. To measure the efficiency of selection, the model must be trained and evaluated for each iteration of the workflow. Lui et al. [8] use an ensemble of models that require training of several models that increase the training time and memory needed. Tong et al. [14] use a genetic algorithm in which individuals are the model, consuming a lot of memory for numerous populations. The computation can be prohibitively expensive with numerous features or large models. For that reason, this approximation is beyond the scope of this paper.

J. Wang et al. [16] and A. Marcano-Cedeño et al. [9] are included in the custom regularization type. In this group of methods, the optimization in the training phase is modified on the basis of some assumptions, such as the distribution of input data. The weights are regularized based on some selected heuristic which can result in suboptimal results. One example is Cancela et al. [2] where a non-convex regularization is used which may difficult the convergence of the model. These types of methods are efficient, but assumptions can affect the convergence of the model. In this paper, any strong assumption is used in the model in order to obtain a general purpose method that does not difficult the convergence of the model in order to minimize the training time.

3 Methodology

3.1 Description

In this section, the formal definition of FADL is explained. Each tensor dimensions are defined as the superindex that excludes the batch dimension.

Let $X^{MxD} \in \mathbb{R}$ be the input taken from the dataset, where M represents the past window size for the time-series and D represents the number of features. Note that in all datasets except time-series the past window size M will always be 1. Let W_L^{1xD} be the weights used in FADL with always 1 past window size and D features. Let H be the Heaviside function that binarizes the input element by element, resulting in a tensor with the same shape as the input.

Formally, FADL can be defined as:

$$FADL(X^{MxD}) = H(W_L^{1xD}) \circ X^{MxD}, \tag{1}$$

where $\circ$ represents the Hadamard product. Moreover, the Heaviside function serves as a gate that indicates if the feature is selected:

$$H(W_L^j) = \begin{cases} 1, \ if \ W_L^j >= 0.5, & (2) \\ 0, \ if \ W_L^j < 0.5, & (3) \end{cases}$$

where W_L^j represents the weight associated to X_j, j $\in$ [1..D], in W_L^{1xD}.

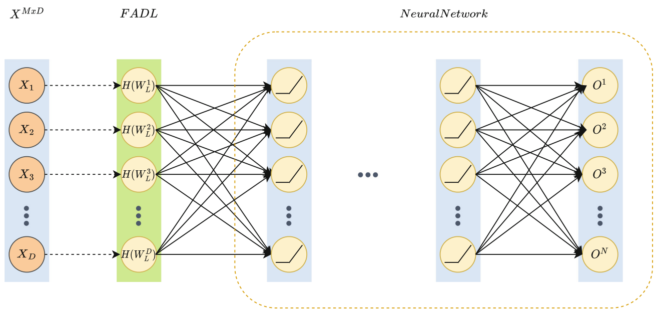

Fig. 1. The proposed FADL with a fully connected neural network ($M = 1$ is assumed). Note that dashed lines represent the element-wise multiplication and solid lines represent the usual fully connected multiplication with weights. Additionally, O_i, $i \in [1, H]$, represents the output of the neural network.

Figure 1 shows FADL applied to a fully connected neural network. Note that the Heaviside function outputs the selection mask, which determines the relevant features of the neural network for the problem. Hadamard product is used between X^{MxD} and $H(W_L^{1xD})$, setting unnecessary features in X^{MxD} to zero.

Backpropagation is used to train FADL as an additional layer in the neural network. When a weight is set to zero, the corresponding feature contribution in the forward step is zero. The feature selection mask is related to the knowledge modeled by the neural network, since the FADL weights are trained as an ordinary layer.

We will study the impact of FADL in two extreme scenarios. On the one hand, if all weights in FADL are positive, FADL would not have any impact on the efficiency of the neural network. On the other hand, if all weights in FADL are negative, that means that the neural network considers all features irrelevant and have no impact on the target.

Note that any of the previous scenarios is desirable. The first scenario does not consider any selection, and FADL serves no purpose other than to notify us that all features are useful for the model. The regularization term described in the following subsection is used to avoid this scenario. The second scenario is not desired as this means that no informative features have been selected. The second situation is uncontrollable since it might be caused by external factors such as a poor data gathering procedure, a poor model configuration, or even the problem is not well modeled using a neural network approach.

3.2 Weight Initialization and Regularization

In FADL, weight initialization may be thought of as the initial hypothesis of the influence of the features. Large positive values introduce a bias, indicating that the features must be present. As a consequence, the model would need more time to change the sign of the feature values. The same logic can be applied to lower negative weight values. In our case, any strong bias is imposed on the initial weights. The weights are set to positive, but close to zero, to facilitate the change from positive to negative weights without extra time. For this reason, the weights were initialized as 0.01.

Weight regularization is another crucial element in FADL that can be thought as a loss tolerance and avoid undesirable scenarios. The regularization used in FADL is Lasso regression over the feature selection mask. Note that we are not interested in regularizing the weights of the FADL, but the feature selection mask result of the Heaviside function. Regularization penalizes the selection of the feature with a constant value independent of the weight value. In this way, the fewer features in the feature selection mask, the higher penalization is added.

Penalization helps avoid previously described undesirable scenarios. We mentioned the extreme scenario in which the feature selection mask selects all features. In this case, the penalization is maximum and would be avoided. The second undesirable scenario is when the n features can explain the target with the same loss as the m features, where $n > m$. Without regularization, any option would be selected, as the loss introduced by both are the same. However, with regularization, the option with fewer features would be preferred.

4 Experimentation

4.1 Datasets

The datasets were chosen from a variety of disciplines where feature selection is important. Healthcare, natural disaster impact, fault detection, and air pollution prediction are just a few examples of these application sectors.

BreastCancer dataset [17] contains 10 features of a digitized image of a fine needle aspirate of a breast mass obtained from clinical cases between 1989 and 1991. The goal is to classify benign and malignant tumour.

Heart dataset [12] contains 19 features obtained from the Behavioral Risk Factor Surveillance System (BRFSS) in U.S. territories through telephonic interviews. The goal is to predict the existence of any coronary heart disease (CHD) or myocardial infarction (MI).

The Cancer Genome Atlas dataset (TCGA) [5] contains 20531 gene expression levels through RNA-Sequencing technique using illumina HiSeq platform. The dataset contains an extraction of gene expression from patients with different types of tumors. The goal is to predict the existence of different types of tumors: Breast invasive carcinoma (BRCA), Kidney renal clear cell carcinoma (KIRC), Colon adenocarcinoma (COAD), Lung adenocarcinoma (LUAD) and Prostate adenocarcinoma (PRAD).

Earthquake dataset [3] contains 38 features collected through surveys by Kathmandu Living Labs and the Central Bureau of Statistics in Nepal. The goal is to predict the level of damage to buildings caused by the 2015 Gorkha earthquake on three levels: low, medium, and high.

WaterPump dataset [13] contains 39 features from waterpoints across Tanzania, collected from Taarifa waterpoints dashboard, which aggregates data from the Tanzania Ministry of Water. The goal is to predict the operating condition of a waterpoint, which can be functional, functional but needs repair, or non-functional.

Torneo dataset [6] contains four pollutants and three meteorological features obtained by sensors in the area of Torneo (Seville) in a hourly basis. The dataset has been divided into four different data sets, each with one pollutant: CO, NO_2, O_3, and PM_{10}.

Table 1 represents the characteristics of each dataset. Except for the features that represent the identifiers, the categorical features were encoded using one-hot encoding. Any missing values in the dataset have been replaced by the most common value for that feature. Except for categorical data, all data has been normalized to improve the convergence of the neural network. Data have been separated into train, valid and test with 70%, 10% and 20% of the data in each dataset, respectively. In the case of all Torneo datasets, the time-series has been transformed to forecast one step ahead for the target, including all features from the past 24 h. In the train/valid/test split, the older records are in the train split, while the new records are in the test split.

Table 1. Number of features, number of instances, sequence size, number of targets, and type of problem for each data set.

	#Features	#Instances	Sequence size	#Targets	Type
BreastCancer [17]	10	570	1	2	Classification
Heart [12]	19	319,796	1	2	Classification
TCGA [5]	20,531	802	1	5	Classification
Earthquake [3]	38	260,602	1	3	Classification
Waterpump [13]	39	59,401	1	3	Classification
TorneoCO [6]	7	4,017	24	1	Regression
TorneoNO_2 [6]	7	4,017	24	1	Regression
TorneoO_3 [6]	7	4,017	24	1	Regression
TorneoPM_{10} [6]	7	4,017	24	1	Regression

4.2 Experimental Settings

The chosen fully-connected model consists of an input layer, an output layer and two hidden layers, one with up to half the number of features and the other with up to a quarter of the number of features. The selected activation function is relu in all hidden layers and, in the output layer, softmax and linear activation function are selected for classification and regression tasks, respectively. The neural network was trained for 100 epochs with Adam optimizer using a learning rate of 0.001 and early stopping with 10 epochs maximum. The model was chosen because it was simple, efficient and all-purpose. The purpose of our paper is to focus on feature selection rather than tuning the hyperparameters of the model.

Our proposed method is compared to four different strategies: no selection (NS), correlation feature selection (Corr), linear feature selection (Linear) and mutual information feature selection (MI). No selection technique is used as a baseline since it does not perform any selection. Correlation feature selection analyzes the relationship between the features and the target in order to provide a score. Linear feature selection requires fitting a linear model and calculating a score to each feature based on the coefficients of the model. Mutual information selection [4], assigns a score to each feature based on its dependence on the target.

As the proposed feature selection methods are parametric, we need to set a threshold to select the relevant features. To make the selection, the features were ranked according to their score. Then, we select the features whose sum of scores is greater than a threshold. The selected threshold is in the range [60%, 95%] in steps of 5%. Ranges below 60% are not considered due to computation limitations. Note that feature selection is done to all time steps in torneo datasets, as feature selection based on time is outside the scope of this paper.

The metrics were separated into two categories: efficacy metrics and performance metrics. The performance metrics measure the number of features selected by the method and time used to select the features plus the time used to train the model. Note that in the no selection strategy and in FADL, the model is trained just once, as there is no threshold optimization. The Mean Squared Error is used as an efficacy metric in regression, while the F1 score is used in classification. As the parametric methods need to optimize the threshold, the result with best efficacy and performance is used for each dataset. However, the parametric feature selection methods need to execute the model several times to obtain the optimal threshold. For that reason, to perform fair performance comparisons between the methods, the sum of times for each threshold is used.

4.3 Results and Discussion

In this section, we analyze our method in terms of both effectiveness and performance perspectives, both for classification and regression tasks. Specifically, Fig. 2 shows the effectiveness achieved by a neural network trained using the proposed FADL, in addition to the rest of benchmark methods, both for classification (F1-Score) and regression (Mean Squared Error) datasets.

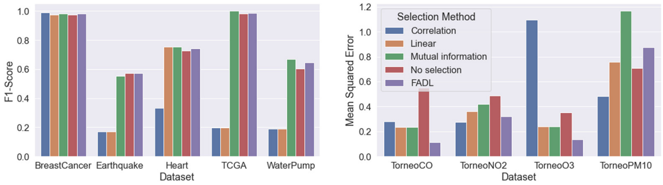

Fig. 2. Efficacy results for classification (F1-Score) and regression (MSE).

In classification datasets, Mutual information method achieves the best efficacy in Heart, TCGA and Waterpump datasets; Correlation method achieves the best efficacy in BreastCancer and FADL method in Earthquake dataset. In general, the best feature selection methods for classification are mutual information and FADL as they obtain good results in almost all datasets. Additionally, mutual information and FADL obtain better results than No selection except in Earthquake dataset where only the FADL obtains the same results as No selection. Note than Correlation and Linear methods obtain poor results in Earthquake, TCGA and WaterPump due to any of the thresholds in the range explored being adequate for such datasets.

In regression datasets, the best selection methods are Correlation and FADL. In TorneoCO and TorneoO_3 datasets, there is a great improvement in FADL compared to other methods. In TorneoNO_2 dataset, the best method is correlation but has little difference with FADL. In TorneoPM_{10} dataset, all selection methods perform poorly except for correlation. In general, most of the methods improve the no selection approach except for TorneoPM_{10} where the selection seems to be harder compared to other regression datasets.

Figure 3 shows the selected features for the best threshold and the sum of training time. Analyzing the number of features, feature selection methods that

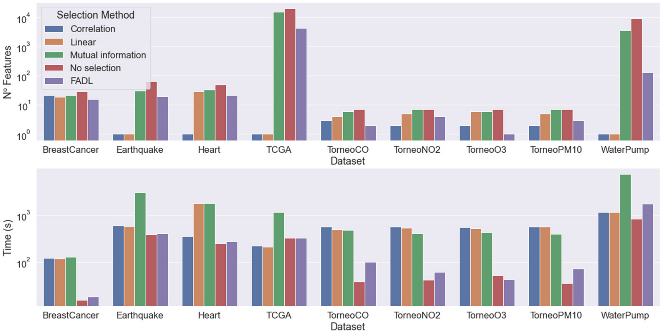

Fig. 3. Number of features and training time for each feature. Note that the y-axis is on a logarithmic scale.

perform poorly in terms of efficiency select just one feature. Apparently, the score function used in each selection method assigns a great value to one feature that represents at least 95% of the total sum of scores. In classification datasets, the FADL is the method which, obtaining good efficacy results in all datasets, selects fewer features than other methods with similar efficacy results. In regression, correlation method obtains fewer features in two datasets and FADL in the remaining regression datasets.

In terms of execution time, all methods obtain an execution time similar to or greater than no selection. The methods that achieved poor efficacy obtained less execution time because the selection was not adequate. In cases where the selection was performed adequately, FADL obtains better execution times than any other selection method. In general, no selection approach obtains fewer times and FADL shows little increment except for regression datasets and Waterpump.

5 Conclusions and Future Works

In this paper, we proposed a nonparametric model-aware feature selection with stable results. The FADL provides interpretability for neural networks, indicating the relevant features used by the model.

The FADL achieved the best execution times, obtaining a competitive selection specially in TorneoCO and TorneoO3 datasets where the efficacy was greatly improved, the number of features reduced, and the execution time was considerably lower than the other selection methods studied. Furthermore, as the results showed, FADL is the most stable method in terms of efficacy and performance metrics.

We propose expanding the interpretability to another dimension as time in future work, revealing not only the significant features but also the relevant time steps. Furthermore, we try to address the problem with features that are important in only a few circumstances, and some features are only significant at particular times. Because there are so few of them, they cause such little error and can be ignored.

Acknowledgements. The authors would like to thank the Spanish Ministry of Science and Innovation for the support under the project PID2020-117954RB and the European Regional Development Fund and Junta de Andalucía for projects PY20-00870 and UPO-138516.

References

1. Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., Kasneci, G.: Deep Neural Networks and Tabular Data: A Survey (2021)
2. Cancela, B., Bolón-Canedo, V., Alonso-Betanzos, A.: E2E-FS: An End-to-End Feature Selection Method for Neural Networks. CoRR (2020)
3. Nepal Earthquake Open Data (2015). http://eq2015.npc.gov.np/#/
4. E-Shannon, C.: A mathematical theory of communication. ACM SIGMOBILE Mob. Comput. Commun. Rev. **5**(1), 3–55 (2001)
5. Fiorini, S.: UCI Gene Expression Cancer RNA-Seq (2016)
6. Gómez-Losada, A., Asencio-Cortés, G., Martínez-Àlvarez, F., Riquelme, J.C.: A novel approach to forecast urban surface-level ozone considering heterogeneous locations and limited information. Environ. Model. Softw. **110**, 52–61 (2018)
7. Khemphila, A., Boonjing, V.: Heart disease classification using neural network and feature selection. In: Proceedings - ICSEng 2011: International Conference on Systems Engineering (2011)
8. Liu, B., Cui, Q., Jiang, T., Ma, S.: A combinational feature selection and ensemble neural network method for classification of gene expression data. BMC Bioinform. **136**, 10 (2004)
9. Marcano-Cedeño, A., Quintanilla-Domínguez, J., Cortina-Januchs, M.G., Andina, D.: Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In: IECON Proceedings (Industrial Electronics Conference) (2010)
10. Miao, J., Niu, L.: A survey on feature selection. Proc. Comput. Sci. **91**, 12 (2016)
11. Monirul Kabir, Md., Monirul Islam, Md., Murase, K.: A new wrapper feature selection approach using neural network. Neurocomputing. **73**(16), 3273–3283 (2010)
12. Pytlak, K.: Personal Key Indicators of Heart Disease (2020). www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease
13. Taarifa: Water pump (2022). https://taarifa.org/
14. Tong, D., Mintram, R.: Genetic algorithm-neural network (GANN): a study of neural network activation functions and depth of genetic algorithm search applied to feature selection. Int. J. Mach. Learn. Cybern. **1**, 75–87 (2010)
15. Verikas, A., Bacauskiene, M.: Feature selection with neural networks. Pattern Recogn. Lett. **23**(11), 1323–1335 (2002)
16. Wang, J., Zhang, H., Wang, J., Pu, Y., Pal, N.R.: Feature selection using a neural network with group lasso regularization and controlled redundancy. IEEE Trans. Neural Netw. Learn. Syst. **32**(3), 1110–1123 (2021)
17. Wolberg, W.H., Nick Street, W., Mangasarian, O.L.: UCI Breast Cancer Wisconsin (Diagnostic) (1995)