Finding components of a good accuracy with XAI !

Benjamin CHAMAND[1], Olivier RISSER-MAROIX[2]

[1]IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, France
[2]LIPADE, Université Paris Cité, France

Abstract

This research presents a pipeline to find the key elements to achieve high accuracy. Indeed, one of the most common tasks in machine learning is classification, and numerous loss functions have been created to maximize this non-differentiable goal. Previous work on loss function design was mainly guided by intuition and theory before being validated by experience. Here, we use a different approach: we aim to learn from experiments. This data-driven method is comparable to how general laws are found from data in physics. We automatically discovered a mathematical expression on more than 260 datasets that is highly correlated with the accuracy of a linear classifier. More interestingly, this formula replicates key findings from several earlier papers on loss design and is highly explainable. We hope this research will open up novel possibilities for developing new heuristics and foster a deeper comprehension of machine learning theory.

Keywords

Symbolic Regression, Explainability, Datasets Representation

1. Introduction

Most machine learning (ML) research involves creating and assessing components based on theoretical intuitions. Acquiring knowledge from experimentation would be a distinct strategy, similar to how physicists have attempted to deduce the analytical laws underlying the physical processes in nature from observations. With the development of AI, a new tendency to automate and support research with ML tools is emerging. Some mathematics [1] and physics [2, 3] researchers started to use it. The most similar approach in machine learning (ML) would be meta-learning, where a model gains experience throughout numerous learning sessions to enhance its performances without human intervention. Although this paradigm has been used successfully for many tasks, including hyperparameters optimization and neural architecture search (NAS), the solutions found are generally not explainable. Thus, it is not so surprising that the use of AI as a tool to assist in theoretical findings in ML research has received so little attention.

Understanding the mathematical relationships between the variables in a given system is a requirement of the scientific method. Symbolic regression (SR) aims to solve the problem of finding a function that explains the hidden relationships in the data without knowing the structure of the function beforehand. Given that SR is NP-hard, evolutionary approaches have been created to find approximations of solutions [4, 5, 6].

While the task of predicting accuracy may look odd at first glance, solving it has multiple applications such as: fastening NAS [7, 8]; evaluating the accuracy of a classifier on an unlabeled test set [9]; or measuring the difficulty of a dataset [10, 11]. Previous works mostly rely on neural networks or random forests, making their solutions found non-explainable [9, 12]. The text classification task has already been studied with features such as *n-grams* [10]. Their approach is nevertheless constrained by choice of features, which limits it to textual datasets, and by only finding an unweighted summation of some of those statistics. Statistics to characterize datasets have been investigated in broader contexts [13, 14, 15]. While studying each variable independently, [14] suggested that the relationship between such statistics and the difficulty of a dataset is complex and would require a nonlinear combination of those variables.

In this work, we propose a pipeline able to produce a general formula predicting the future performance of a linear classifier with a strong Pearson's correlation and r^2 score. We found our solution highly explainable and examined it in the context of decades of research.

2. Proposed Approach

Datasets and Feature Extractors We choose 12 datasets and 22 feature extractors using the same manner as [13] to find a general law spanning a large range of factors for a classification challenge. The amount of classes varies from 10 to 1854, and the dimension of the embeddings spans from 256 to 2048. We used datasets such as CIFAR10, CUB200, ImageNetMini, or THINGS. To cover a large number of dimensions and difficulty levels of linear classification, varied architectures with different pretraining have been chosen. Some of them are kept untrained. We used different variants of popular feature extractors such as ResNet, MobileNet, SqueezeNet,

Proceedings of the CIKM 2022 Workshops, October, 2022, Atlanta, USA
benjamin.chamand@irit.fr (B. CHAMAND);
orissermaroix@gmail.com (O. RISSER-MAROIX)

CLIP, etc. We construct a meta-dataset $\mathscr{M}$ from those 264 datasets of embeddings (the combination of all datasets by all feature extractors).

Meta-Dataset Representation To be able to find the hidden relationship between a given dataset and the associated optimal accuracy, we need to describe each of those datasets by a feature vector s in a shared representation space $\mathscr{S}$. We crafted 19 features s_i such as: the dimensionality of embeddings (*dim*), the number of output classes (*n_classes*), the traces of the average matrices of all intra-class and inter-classes covariance matrices (*sb_trace*, *sw_trace*), the mean cosine similarity between each pair of dimensions (*feats_cos_sim*), the cosine similarity between prototypes (*prototype_cos_sim*), etc.

Ground Truth Creation After extracting the embeddings from diverse datasets using feature extractors, we need to determine the best achievable accuracy via a softmax classifier for each dataset of embeddings. We divided each embedding dataset into testing and training sets and learned the model for 1000 epochs with a batch size of 2048. As pre-processing, all embeddings were only ℓ_2-normalized. By tracking the accuracy on the test set, we can observe the best-reached accuracy α, an approximation of the best accuracy reachable α^*. Our meta-dataset $\mathscr{M} = \{(s_i, \alpha_i)\}_{i=1}^{D}$ corresponds to all the pairs of statistical representation $s_i \in \mathscr{S}$ of each dataset d_i of the D datasets and the observed optimal accuracy $\alpha_i \in \mathscr{A}$. These tuples contain our inputs and outputs.

Symbolic Regression We use the *gplearn* implementation because of the compactness of the solutions, speed of execution, robustness to noise [16], and ease of use. The set of primitive function used is $\{\log, e, \sqrt{}, +, -, \times, \div\}$ and the set of terminals corresponds to the statistics s_i describing the dataset d_i. We evolved a population of 5000 individuals for 20 steps. We designed a fitness function $\mathscr{F}$ such that both: pretrained and untrained extracted embeddings have a linear correlation with accuracy, independently. We split our meta-dataset in a fixed 75/25-train/test fashion and repeat each experiment 1000×. Since $\mathscr{F}$ only seeks for correlation, a linear transformation of the output value is learned on the training set in order to predict the accuracy ($\hat{\alpha} = a \cdot p(\cdot) + b$).

3. Results

Baselines To evaluate the performance of our GP solution, we compare it with popular regression methods using the same train/test split. Performances on the test set are reported in Table. 1. The substantial gap of r^2 score between the linear regressor suggests that the task of predicting the accuracy requires a nonlinear combination of variables. Thus, we compare nonlinear regressors such as decision trees and random forests because of their performances and the widespread belief that those models are among the most interpretable ones. Our formula outperformed them while being more explainable.

Table 1
Our formula has a better correlation and higher predictive power with only 5 variables (all p-value < 0.01).

Method	Pearson$r$	r^2
Linear Regression	0.9042	0.8011
Decision Tree Regressor	0.9472	0.8868
Random Forest Regr. (10 trees)	0.9643	0.9246
Our GP formula (*GPF*)	**0.9682**	**0.9319**

Symbolic Regression Formula We ran our GP pipeline 1000× on the same training set and serialized their respective solutions and scores for analysis. The solution having the best test r^2 score was found 6×. Our formula has a complexity of 6 nodes. We will refer to this ***G**enetic **P**rogramming **F**ormulas* as:

$$GPF = \log\left(\frac{sb\_trace/st\_trace}{\sqrt{n\_classes \cdot feats\_corr \cdot prototypes\_cos\_sim}}\right) \tag{1}$$

We can easily rewrite : $GPF = SEP - COR$ with:

$$SEP = \log\left(\frac{sb\_trace}{st\_trace}\right)$$
$$COR = \frac{1}{2}\log\left(n\_classes \cdot feats\_corr \cdot prototypes\_cos\_sim\right) \tag{2}$$

SEP may correspond to a **sep**arability criterion while *COR* may correspond to **cor**relation information. We found those two parts to be complementary. Indeed *SEP* and *COR* have respectively a pearson$r$ of only 0.65 and −0.87. Finally, we found that other best-performing GP formulas have similar structures and variables.

4. Discussion

GPF can be written as a summation of two components. One can see that the first element *SEP* is close to the Fisher's criterion used in the *Linear Discriminant Analysis* (LDA) [17] where the objective is to find a linear projection that maximizes the ratio of between-class variance and the within-class variance. Thus, *SEP* corresponds to a separability measure of classes. Remarkably, this criterion has been effectively applied as a loss function in deep learning [18, 19]. The choice of an LDA-based loss function remains marginal in deep learning, the cross-entropy (CE) being a more popular choice. However, strong similarities between the LDA and the CE allow us to swap this first separability measure with the latter one. Indeed, [20] noticed that one of the most widely studied technical routes for the CE-based losses is to encourage stronger intra-class compactness and larger inter-class separability such as the Fisher's criterion. The second part, *COR*, is negatively correlated to the accuracy. The first variable is the number of classes (*n_classes*). Indeed, it is natural to expect scores to decrease as the number of classes grows. For example [21] observed a drop in accuracy on the CUB200 dataset when changing the number of classes from a coarse level to a fine-grained one.

In defense of the weights decorrelation term (*prototypes_cos_sim*), [22] found on several state-of-the-art CNN that they could achieve better accuracy, more stable training, and smoother convergence by using orthogonal regularization of weights. Previous works on features decorrelation heavily justify the presence of our features decorrelation variable (*feats_corr*) [23, 24, 25, 20, 26]. Indeed, [25] found that correlated input variables usually lead to slower convergence. Thus several propositions were developed to better decorrelate variables such as PCA, or ZCA. More recently, decorrelation played an important role in the performance increase of self-supervised methods [23, 24, 26].

In this paper, we showed that a simple pipeline could help us to extract theoretical intuitions from experimentation. Our formula is highly explainable and is consistent with decades of research. While this work is still ongoing, we are working on an extended version [27].

References

[1] A. Davies, et al., Advancing mathematics by guiding human intuition with ai, Nature 600 (2021) 70–74.

[2] M. R. Douglas, Machine learning as a tool in theoretical science, Nature Reviews Physics (2022).

[3] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (2009).

[4] D. A. Augusto, H. J. C. Barbosa, Symbolic regression via genetic programming, in: SBRN, Procs., 2000.

[5] J. R. Koza, Genetic programming - on the programming of computers by means of natural selection, Complex adaptive systems, MIT Press, 1993.

[6] Q. Lu, J. Ren, Z. Wang, Using genetic programming with prior formula knowledge to solve symbolic regression problem, Computational Intelligence and Neuroscience (2016).

[7] R. Istrate, et al., Tapas: Train-less accuracy predictor for architecture search, in: AAAI Procs, 2019.

[8] W. Wen, et al., Neural predictor for neural architecture search, in: ECCV, Procs., 2020.

[9] W. Deng, L. Zheng, Are labels always necessary for classifier accuracy evaluation?, in: CVPR, Procs., 2021, pp. 15069–15078.

[10] E. Collins, N. Rozanov, B. Zhang, Evolutionary data measures: Understanding the difficulty of text classification tasks, in: CoNLL, 2018.

[11] F. Scheidegger, R. Istrate, G. Mariani, L. Benini, C. Bekas, C. Malossi, Efficient image dataset classification difficulty estimation for predicting deep-learning accuracy, The Visual Computer 37 (2021).

[12] Y. Yamada, T. Morimura, Weight features for predicting future model performance of deep neural networks., in: IJCAI, 2016, pp. 2231–2237.

[13] B. Chamand, O. Risser-Maroix, C. Kurtz, P. Joly, N. Loménie, Fine-tune your classifier: Finding correlations with temperature, in: ICIP, Procs., 2022.

[14] T. K. Ho, M. Basu, Complexity measures of supervised classification problems, TPAMI 24 (2002).

[15] A. C. Lorena, L. P. Garcia, J. Lehmann, M. C. Souto, T. K. Ho, How complex is your classification problem? a survey on measuring classification complexity, ACM Computing Surveys 52 (2019).

[16] W. La Cava, P. Orzechowski, B. Burlacu, F. O. de França, M. Virgolin, Y. Jin, M. Kommenda, J. H. Moore, Contemporary symbolic regression methods and their relative performance, in: J. Vanschoren, S. Yeung (Eds.), Proceedings of the NIPS Track on Datasets and Benchmarks, 2021.

[17] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of eugenics 7 (1936).

[18] M. Dorfer, R. Kelz, G. Widmer, Deep linear discriminant analysis, in: ICLR, Procs., 2016.

[19] B. Ghojogh, et al., Fisher discriminant triplet and contrastive losses for training siamese networks, in: IJCNN Procs, IEEE, 2020.

[20] W. Wan, Y. Zhong, T. Li, J. Chen, Rethinking feature distribution for loss functions in image classification, in: CVPR, Procs., 2018.

[21] D. Chang, K. Pang, Y. Zheng, Z. Ma, Y.-Z. Song, J. Guo, Your" flamingo" is my" bird": Fine-grained, or not, in: CVPR, Procs., 2021.

[22] N. Bansal, X. Chen, Z. Wang, Can we gain more from orthogonality regularizations in training deep networks?, in: NIPS, Procs., 2018.

[23] A. Ermolov, A. Siarohin, E. Sangineto, N. Sebe, Whitening for self-supervised representation learning, in: ICML, Procs., PMLR, 2021.

[24] T. Hua, et al., On feature decorrelation in self-supervised learning, in: ICCV, Procs., 2021.

[25] Y. A. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient BackProp, 2012.

[26] S. Zhang, et al., Zero-cl: Instance and feature decorrelation for negative-free symmetric contrastive learning, in: ICLR, Procs., 2022.

[27] O. Risser-Maroix, B. Chamand, What can we learn by predicting accuracy?, 2022. arXiv:2208.01358.