




Learning Curves Prediction for a Transformers-Based Model

Francisco Cruz ¹, Mauro Castelli ^{1*} 

¹ NOVA Information Management School (NOVA IMS), Universidade NOVA de Lisboa, Campus de Campolide, 1070-312 Lisboa, Portugal.

Abstract

One of the main challenges when training or fine-tuning a machine learning model concerns the number of observations necessary to achieve satisfactory performance. While, in general, more training observations result in a better-performing model, collecting more data can be time-consuming, expensive, or even impossible. For this reason, investigating the relationship between the dataset's size and the performance of a machine learning model is fundamental to deciding, with a certain likelihood, the minimum number of observations that are necessary to ensure a satisfactory-performing model is obtained as a result of the training process. The learning curve represents the relationship between the dataset's size and the performance of the model and is especially useful when choosing a model for a specific task or planning the annotation work of a dataset. Thus, the purpose of this paper is to find the functions that best fit the learning curves of a Transformers-based model (LayoutLM) when fine-tuned to extract information from invoices. Two new datasets of invoices are made available for such a task. Combined with a third dataset already available online, 22 sub-datasets are defined, and their learning curves are plotted based on cross-validation results. The functions are fit using a non-linear least squares technique. The results show that both a bi-asymptotic and a Morgan-Mercer-Flodin function fit the learning curves extremely well. Also, an empirical relation is presented to predict the learning curve from a single parameter that may be easily obtained in the early stage of the annotation process.

Keywords:

Dataset Size;
Document Data Extraction;
Fine-Tuning;
Learning Curves;
Transformers.

Article History:

Received: 06 June 2023
Revised: 08 September 2023
Accepted: 19 September 2023
Published: 01 October 2023

1- Introduction

Machine learning (ML) models have been applied to address problems over different domains [1] and represent a fundamental tool for solving problems in which traditional statistical methods cannot be used. However, despite its popularity, ML is mainly guided by empirical experience rather than theory. For this reason, several fundamental choices that are critical to obtaining a good-performing ML model represent a challenge in the ML pipeline. For instance, given an ML technique, one of the most challenging tasks is choosing the hyperparameters' values [2]. Despite the vast ML literature, there are no formal rules for determining, given an optimization problem and an ML technique, the best parameters that will result in the best performance of the ML model. For instance, focusing on artificial neural networks (ANNs), there are no formal rules to design the topology in such a way as to maximize the performance of the ANN on a given task.

Another relevant issue that ML practitioners must face concerns the number of observations that are necessary for achieving a good-performing model as a result of the training process. This task is particularly important, even in an epoch characterized by the availability of vast amounts of data. In fact, training an ML model using all the available data may require an unbearable amount of time and computational resources, and the performance improvement resulting from training a model with more data is, at a given point, negligible.

* **CONTACT:** mcastelli@novaims.unl.pt

DOI: <http://dx.doi.org/10.28991/ESJ-2023-07-05-03>

© 2023 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Thus, when dealing with vast amounts of data, a relevant issue is determining the minimum number of observations necessary to train a model in a way that can guarantee a satisfactory performance. On the other hand, determining the minimum number of training observations is an even more relevant challenge when focusing on problems characterized by the availability of a limited number of observations. For instance, in some domains, collecting data is a time-consuming and expensive task, and, in some cases, collecting additional training observations is simply impossible. Thus, in such a situation, being able to decide whether the available observations are enough for a successful training process would reduce the cost associated with gathering additional observations.

Despite the importance of this problem, the ML literature does not present significant contributions in this area [3], and, usually, the experience of ML practitioners and empirical methods guide the choice concerning the size of the training set. For instance, it is generally expected that the prediction skills of machine learning models improve with the amount and quality of data available for training. Nevertheless, it is also expected that, independently of data quality, there is likely to be a non-zero lower-bound error past which models will be unable to improve [4]. Thus, if there is a limit above which adding extra data to train models brings small or no gains, what is that limit? The answer to this question is especially useful to have an estimation of how many samples need to be annotated when training a model using a new dataset.

As previously pointed out, a limited number of studies have been proposed in the literature for very specific applications. In Beleites et al. [5], the authors investigated the performance of an ML model as a function of the training sample size in bio spectroscopy classification tasks. In particular, they analyzed the tradeoff between performance and training set size for small sample size cases, with up to 25 observations per class. The authors show that around 100 samples are usually needed to achieve good performance on the considered problems and ensure good performance on unseen data. While the study is interesting for the specific application, the findings cannot be generalized to other benchmarks and tasks. Another study dealing with classification tasks in bioinformatics was proposed by Dobbin and Simon [6]. They developed a method to determine the size of the training set for a prediction task in the context of high-dimensional data (i.e., where each observation consists of a significant number of variables). The method is based on a parametric probability model, and the authors showed that many prediction problems do not require a large training set for building a classifier with satisfactory performance. Similarly, Dobbin et al. [7] investigated training set size for the task of building a classifier for gene expression microarray data. They presented a model-based approach to determining the sample size required to adequately train a classifier. Based on the experimental results, they concluded that sample size can be determined from three quantities: standardized fold change, class prevalence, and the number of genes or features on the arrays. Based on these findings, they developed a tool for the prediction of dataset size for building gene expression classifiers.

A more general study investigated the influence of the training data on the classification error of multiclass classifiers. To make a concrete statement, the authors focused on the k-NN classifier because of its large adoption in industrial settings. The proposed method is tested on four different multiclass problems, showing different qualities in terms of training set size estimation [8]. Despite the existence of some works, the problem is still poorly investigated in the ML literature. The existing studies focused on simple ML techniques, and, to the best of our knowledge, no studies investigated the effect of the training set size on more advanced and state-of-the-art deep learning models. To answer this call and fill the existing gap in the ML literature, in this study, we investigate the effect of fine-tuning the training set size on the performance of a transformer-based model [9].

The work stems from the importance of the relationship between dataset size and model quality. Having access to such a relationship would help practitioners estimate how much data they would need to collect and annotate to obtain satisfactory modeling results. The relation between the dataset size and the model performance (or error) may be given by a graphical representation, which in this work, following existing literature [10], is called the learning curve. It should be noted that “learning curve” is a term widely applied, often with different meanings. For example, the learning curve considered in this work is different from the curve that displays the value of any objective function as a function of the number of epochs or iterations used for optimization.

For every new batch of samples made available, the existing dataset may be trained and evaluated, and the dataset size may be plotted against a performance metric (ex: the F1 score or the error rate). Such a curve reflects the concept of learning curves used not only in this work but also throughout the literature [10]. This representation is extremely useful to understand not only the learning pace of a model for a given dataset but also to understand whether there is a point after which adding more samples to the dataset is useless to improve the model performance. Also, extrapolating the learning curve may give a preliminary indication of how many examples to collect to achieve a specific performance, thus allowing one to judge when data collection can be stopped [10].

Another essential keyword in this work is transfer learning, which involves knowledge transfer across domains or tasks. It challenges the common assumption that both training and test data should be drawn from the same distribution [11]. As the usage of learning curves is transversal to any kind of model that requires data, it is also applicable in the fine-tuning stage of a pre-trained model.

Most of the research studies on learning curves focus on two specific subjects: finding a function that fits learning curves and predicting the parameters of such a function. Learning curves for machine learning models such as decision trees [12], SVMs, or KNNs [13] generally show a very good fit for a power law. Moreover, exponential [14] and logarithmic [15] functions may also be a good fit for such models. As for deep learning models, studies mostly claim to find the best fit for power-law behavior [10]. It is important to highlight that such a relationship is considered empirical and is yet to be explained by theoretical work [4].

Also, the procedures to plot a learning curve on the existing research works are not strict. Some evaluate the model performance using holdout, having a fixed test set that is independent of the training dataset size [16–18]. Other researchers use cross-validation, in which the k-fold size increases with the dataset size [12–14, 19, 20].

A few works were found focusing on what determines the parameters of the fit functions. It was found that the asymptotic value of the power law and its exponent could be related [18, 21]. Mukherjee et al. [22] construct empirical learning curves for several molecular classification problems. Yet, most studies on learning curves for deep learning models consider the learning phase of a deep learning model trained from scratch and not the fine-tuning phase. One of the few works that analyze learning curves for the fine-tuning stage is presented by Hoiem et al. [21], who focus specifically on analyzing the learning curve to compare performance between models and not on the prediction of the curve itself.

Considering the problem of having a better knowledge of learning curves and the reduced literature about it, two objectives are defined. First, to predict which functions would better fit the learning curves for a specific transformer-based model (LayoutLM), data originated from three datasets of invoices. Second, to define an empirical formula to estimate the learning curve parameters when only a few samples are available.

Knowing the learning curve of a model with none or few data available may provide a useful estimation of the minimum size of a dataset needed to obtain consistent results after training. This depends on several factors, such as dataset quality, type of data used, model algorithm and hyperparameters, as well as training techniques. Consequently, this work is considered an empirical approach to the problem. Nevertheless, it may be used as a solid reference by authors or users who wish to use the same model to extract information from similar sources (invoices). Furthermore, it is also considered a valid and useful contribution to, as advocated by other authors, make empiricism on data mainstream [23, 24] and promote the usage of learning curves as part of a standard learning system evaluation [21]. One example of how such empiricism has gained importance in the practical usage of deep learning models is the Model Cards of the well-known Hugging Face repository (<https://huggingface.co/docs/hub/model-cards>). Such a tool allows the users to easily see how a model has performed when trained on a wide variety of datasets.

Additionally, two new datasets with invoices, including annotation of relevant fields, are made available. Those were gathered, curated, and pre-processed by the authors for this specific research work. Yet, such datasets will certainly be useful to other researchers or users who want to apply supervised learning to document data extraction. All in all, the contributions of the paper are the following:

- A method to estimate the size of the dataset necessary for fine-tuning a transformer-based architecture with a certain performance;
- The availability of two new datasets that may be used by ML practitioners interested in computer vision tasks and, in particular, the automatic recognition of relevant information from invoices;
- A simple and interpretable formula for estimating the necessary size of a training set. The formula can be used in different settings and will provide a baseline for future studies aiming at analyzing the tradeoff between the model's performance and the dataset's size.

In the subsequent sections, the used datasets and models are described, as well as the methods to create sub-datasets and plot the learning curves. A brief explanation concerning which functions fit the curves and the empirical relation to define the function parameters is also provided. Then, the results are presented and discussed. In the final section of the manuscript, we summarize the main findings of this work and suggest possible future research avenues.

2- Material and Methods

2-1-Datasets

All the samples used in this work are digitalized invoices or receipts, being provided from 3 different origins. For the sake of clarity, the three original datasets are named FLH, SROIE, and OWN. Each comprises a set of pictures of invoices, having each document its corresponding annotation file with the text transcription of relevant fields.

The first original dataset is named after the owner of the invoices, which is Feels Like Home, Mediação Imobiliária (FLH), a Portuguese company in the tourism industry that manages apartments for short renting. Every document represents a purchase made by FLH, for a total of 813 samples. The second dataset is from the SROIE (Scanned Receipts OCR and Information Extraction) competition [25], including 727 documents. Finally, the OWN dataset [26], which is also original, includes 190 invoices that belong to one of the authors of the paper.

All original datasets include the following annotated fields: merchant name, merchant address, total amount, and date of the invoice. Moreover, FLH and OWN datasets include extra annotated fields: invoice number, tax identification number of the merchant, tax identification number of the buyer, and VAT amount. Both FLH and OWN datasets have a considerable part of low-quality and noisy pictures. Also, the language of the invoices is English for the SROIE dataset while Portuguese for the other two datasets. Such factors are thought to increase the heterogeneity of data, therefore enriching the present analysis.

To increase the diversity of learning curves to analyze, 19 sub-datasets are created based on both variations and combinations of the 3 original datasets. While such sub-datasets are just parts, unions, or transformations of the original datasets, it is thought that they provide a wider variety of learning curves to study. The combinations considered in this study are as follows:

- #1, #2, and #3 represent each of the original datasets;
- #4, #5, and #6 each represent the union of two original datasets;
- #7 represents the union of the three original datasets;
- #8, #9, and #10 represent each of the original datasets, only modeled for the fields “merchant name” and “merchant address”;
- #11, #12, and #13 represent each of the original datasets, only modeled for the fields “total amount” and “invoice date”;
- #14, and #15 represent each of the original FLH and OWN datasets, only modelled for the fields “VAT amount”, “buyer tax id”, “merchant tax id” and “invoice number”;
- #16, #17, and #18 represent each of the original datasets with a specific transformation: after performing OCR on the pictures, the text is sorted reversely;
- #19, and #20 represent each of the original FLH and OWN datasets, selecting only merchant names which start with letters A-L;
- #21, and #22 represent each of the original FLH and OWN datasets, selecting only merchant names which start with letters L-Z.

2-2-Model

LayoutLM [27] is a Transformers-based model which jointly models interactions between text and layout information across scanned document images. LayoutLM has demonstrated state-of-the-art results across several document image understanding tasks, including the extraction of relevant information from scanned documents. For this reason, it represents the ideal model to be considered in the experimental phase. LayoutLM enhances the performance of deep neural networks as it overcomes two main limitations of existing models. In particular, existing models are strongly dependent on human-labeled training samples and do not exploit the availability of vast amounts of unlabeled training samples. Second, they used to leverage pretrained computer vision models but without considering joint training of textual and layout information.

To overcome these limitations, LayoutLM represents input textual information by text embeddings and position embeddings and relies on two additional types of input embeddings: a two-dimensional position embedding representing the relative position of a token within a document and an image embedding for representing scanned token images within a document. While the model does not process images directly, a pre-processing phase performs OCR converting the images to their text and corresponding bounding box coordinates. Then, text and coordinates are tokenized to provide input for the model. A pre-trained model is made available so it may be fine-tuned for specific tasks. Similar to BERT, LayoutLM is fine-tuned by initializing with the pre-trained weights, plugging in a specific model for the downstream task, and training the parameters end-to-end using labeled data. This way, weights are obtained both from unlabeled and labeled data in a semi-supervised [28] manner.

To avoid heavier computations, the pre-trained model chosen is LayoutLM Base, with 110 million parameters. The model hyper-parameters are mostly based on the ones presented in the LayoutLM presentation article [27]. After running a brief parametrization, the original parameters are kept, except for the number of attention heads which was modified from 12 to 8. The choice of maintaining the original set is motivated by the result achieved after a preliminary tuning phase, in which we tried to modify some of the existing parameters' values. In particular, we did not observe a performance improvement after changing the parameters' values with respect to the ones used in the LayoutLM paper. Thus, we decided to maintain the original set of parameters' values, and we only modified the number of attention heads from 12 to 8 as competitive performance can be achieved by also reducing the computational effort.

3- Experimental Settings

This section reports the experimental settings used in the experimental campaign. Figure 1 summarizes all the steps of the experimental pipeline, from the data collection and preprocessing to the determination of the formula to estimate the size of the dataset.

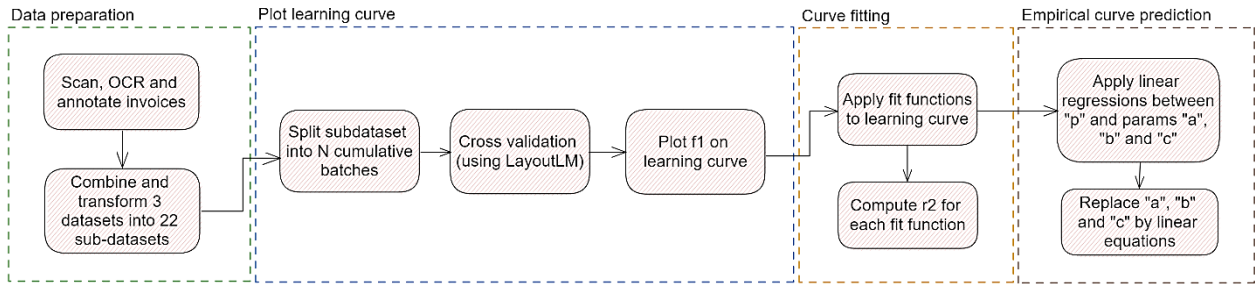


Figure 1. Flowchart reporting the steps performed in this study

The data preparation steps comprised gathering the documents, scanning, and preprocessing, as well as the definition of 22 sub-datasets, as described in the previous section. To plot the learning curves, the first 200 samples of each sub-dataset are split into batches of 5, while the remaining samples are split into batches of 50. The increase of batch size from 5 to 50 is done only to reduce computing time once the learning curve becomes almost flat after around 200 samples for the used sub-datasets. This iterative process correctly simulates the real use case of training and testing while batches of annotated samples become available. For every iteration, the cumulative batch is modeled through 5-fold cross-validation, thus ensuring the robustness of our results and findings. F1 was used to assess the results, and we resorted to the Python package Seqeval [29] for computing the score. Considering that the classes are equally important, the option “micro” is used to average the score over the classes. Thus, the evaluation is made at the word level. F1 is computed for a multi-class problem, yet it excludes the correct predictions on the class “other”, which represents the tokens that are not labeled as relevant fields. The relation between sub-dataset batch size and F1 score is plotted into the learning curve graph. Consequently, 22 learning curves are obtained, one for each sub-dataset.

Given the shape of the learning curves, three functions of different families are chosen as possible candidates for fitting. These are the Morgan-Mercer-Flodin function (MMF4) function (Equation 1), the Hyperbolic function (Equation 2), and the Bi-asymptotic function (Equation 3).

$$f(x, a, b, c, d) = \frac{a \times b + c \times x^d}{b + x^d} \quad (1)$$

$$f(x, a, b, c, d) = b1 + e - c - d(x - a) \quad (2)$$

$$f(x, a, b, c) = \frac{-a}{x-b} + c \quad (3)$$

The three functions are fitted to each of the learning curves through a non-linear least squares technique, using the *optimization.curve_fit method* of the SciPy package for Python [30]. It should be noted that the fit is not applied to every point of the learning curve. For very small sizes of the sub-dataset, it is expected that the model is not able to learn enough to provide scores higher than 0%. Consequently, the learning curve develops only after a specific threshold, having usually an initial straight line with a 0% score. Figure 2 illustrates this 0-score initial stage of the learning curve, with a length equal to p . The fitting of the curve would expectably be affected by the initial stage, so such points are removed before applying the least squares technique. To evaluate the fit, the r2 score is applied by employing the method *metrics.r2_score* of the package Scikit learn [31].

Finally, a separate fit is applied to find a relation between p and one of the fit function parameters. This prediction is made by applying a separate linear regression to the relation between the initial stage size p (see Figure 2) and each of the parameters a , b , and c of the fit function $f(x, a, b, c, \dots)$. Such relation between parameters is then represented by a new function, in which the learning curve may be represented by the parameter p .

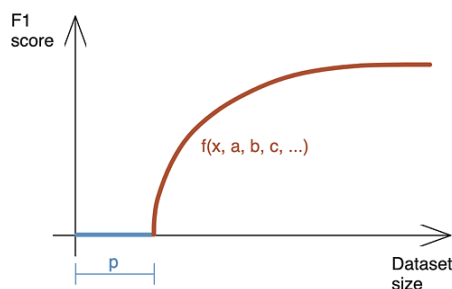


Figure 2. Learning curve with an initial stage with 0% score

4- Results and Discussion

4-1- Curve Fitting

The learning curves for each of the sub-datasets are displayed in Figure 3. It is noticeable how the curves present a similar shape, even though the maximum F1 score varies between $\sim 0,78$ and $\sim 0,99$. The presence of the 0-score initial stage is also evident for most curves. It is confirmed that a model such as LayoutLM is not suited to zero-shot learning when applied to data extracted from invoices, as in most cases, it returns a near 0% score when just a few samples are used for fine-tuning.

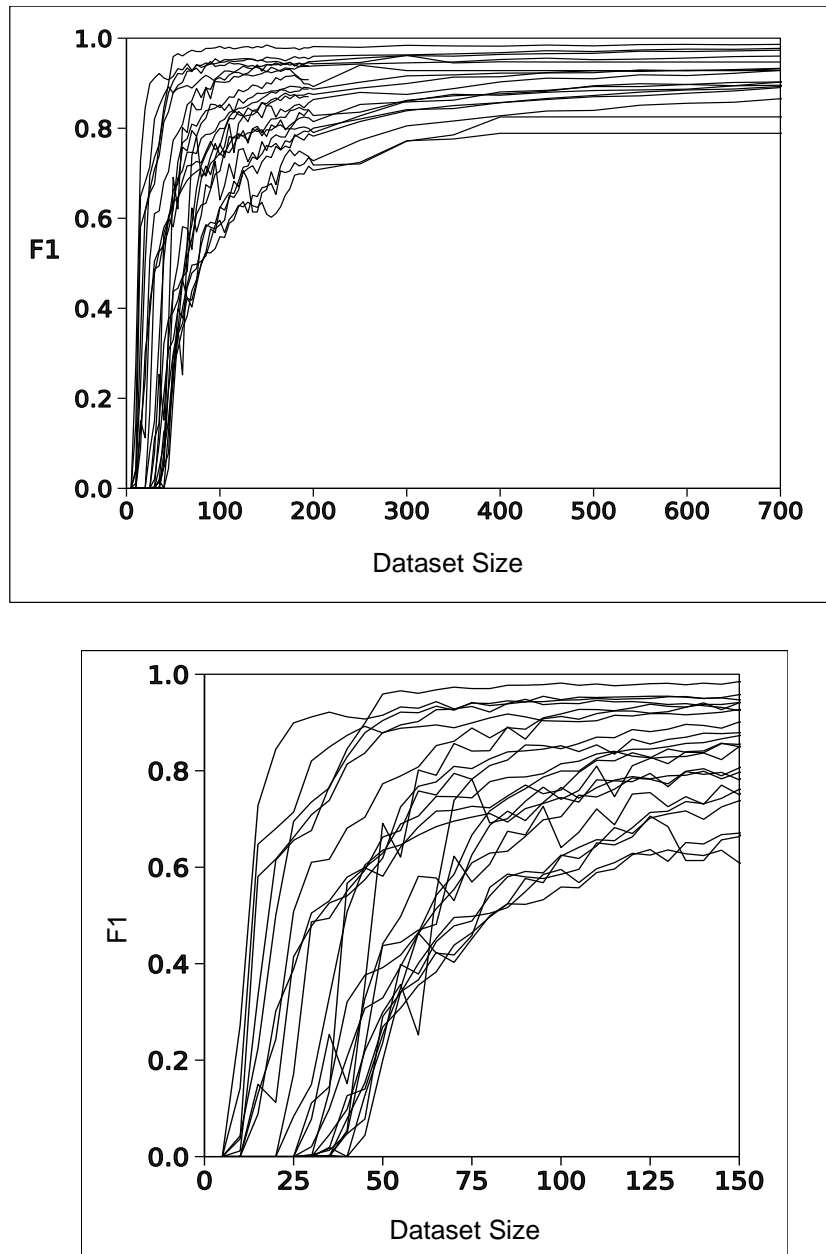


Figure 3. Learning curves for all sub-datasets. Detail on the right side

It should also be highlighted that a relation is visible between the value of p and the maximum value of the curve. The largest is p , the lower is the learning curve performance and its maximum score. Such a relation is explored further to empirically predict a learning curve for each sub-dataset based on the value of p .

Figure 4 shows the boxplots for the r^2 scores obtained for the best fit of each function to each sub-dataset. Both MMF4 and Bi-asymptotic functions present the best fit, with a median of 0,977 and 0,978, respectively. The Hyperbolic function has a slightly lower but still very good performance, with a median r^2 of 0,938.

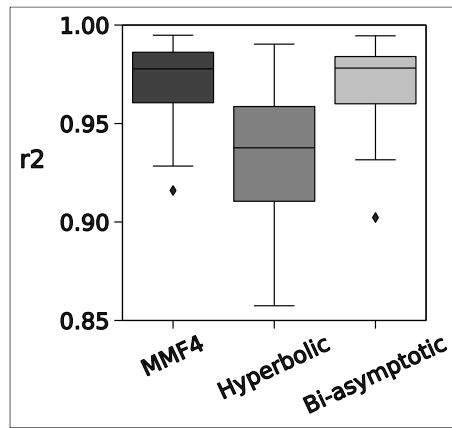


Figure 4. Distribution of r2 scores for each fit function

The results show that, for this specific model and type of data (invoices), the shape of the learning curves may be represented by any of the presented functions: MMF4, hyperbolic, or bi-asymptotic. The fit for any of the functions may be considered excellent, and future works that use the same model and similar data may expect similar shapes for the learning curve.

4-2- Curve Prediction

Regarding the second objective of this work, a relation between the size p of the 0-score initial stage and the curve parameters is explored. The Bi-asymptotic function is used to study this relation, as it presents an excellent fit and has only three parameters. Figure 5 presents the best relation between each of the parameters a , b , c , and p , for the 22 bi-asymptotic functions obtained in the previous curve fit.

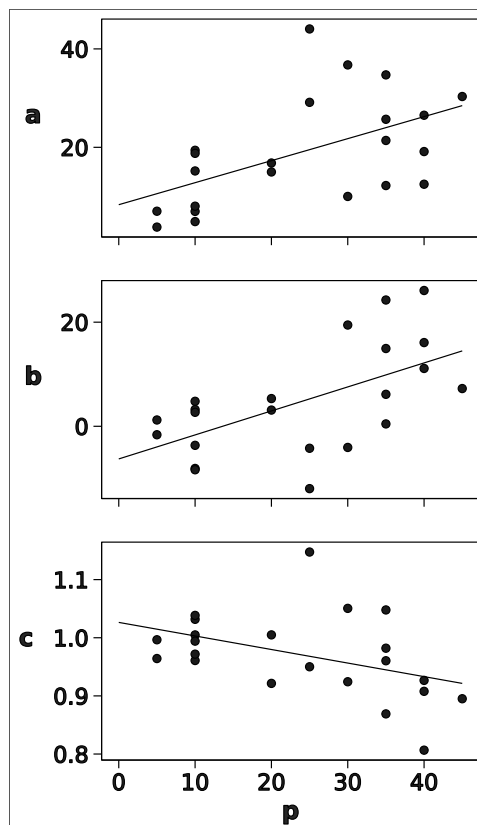


Figure 5. Best linear fit for relations between function parameters a , b , c , and p

The linear functions obtained in the fit are:

$$a = 0.447p + 8.341 \tag{4}$$

$$b = 0.461p - 6.271 \tag{5}$$

$$c = -0.0023p + 1.026 \tag{6}$$

By replacing the parameters a , b , c of Equation 1 with the relations presented on Equations 4, 5 and 6, respectively, the following empirical relation is obtained:

$$f(x, p) = \frac{-0.447p - 8.341}{x + 6.271 - 0.461p} - 0.0023p + 1.026 \quad (7)$$

Figure 6 shows the boxplot with the distribution of r^2 computed for the empirical curves for the 22 sub-datasets. Expectedly, the r^2 median value of 0,86 is lower than the ones obtained with the least squares fit in the previous sub-section. Nevertheless, it would still be considered a good result for an empirical method that relates the learning curve function with only the parameter p .

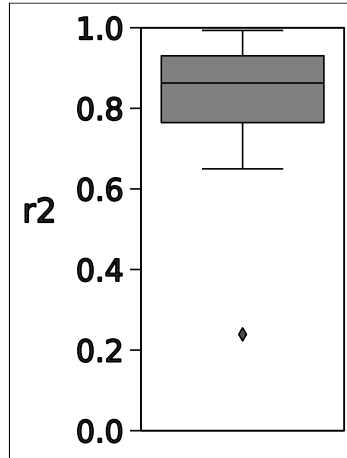


Figure 6. Distribution of r^2 scores for the empirical relation

Moreover, the relation between the bi-asymptotic function parameters and the size p (of the 0-score initial stage) represents a suitable empirical approach to estimating, with very few samples, the learning curve. This would certainly provide a good prediction on the number of samples needed for training and on the model's expected performance.

The main limitation of this work resides in its generalization. Although it was proven that the presented relations are valid for similar applications in which LayoutLM is used to extract data for invoices, this is an empirical method that needs prior confirmation before being used in other use cases. It is expected that when using different models, parameters, and datasets, other mathematical functions may be more suitable to fit the learning curves. Thus, we expect that this research will foster future research concerning the definition of a similar approach to different combinations of models and datasets.

5- Conclusion

This study focuses on finding the functions that best fit the learning curves of a model, as well as developing an empirical relationship between the initial shape of the curve and its parameters. Although empiricism of data has been encouraged by different authors, as well as the importance of learning curves to evaluate machine learning projects, a few works were found in the literature that analyzed learning curves. Within such works, only a few contributions analyzed learning curves on the fine-tuning stage of pre-trained models, none being applied to a similar use case.

To answer this call, the paper presents three datasets, two of which are new to the research community, and the learning curves obtained by applying a pre-trained model (LayoutLM) to 22 combinations of the datasets. Then, three functions are fitted to the learning curves, and an empirical relation is found between the function parameters and the size of the initial stage of the curve. The presented results show an excellent fit of the three functions to the learning curves, as well as a good fit for the empirical relation.

The two research objectives are achieved, and answers may be given. First, the hyperbolic and by-asymptotic functions show the best fit for the learning curves. Second, the size p for which the model shows a near-0% performance can be easily related to the bi-asymptotic function parameters, thus allowing the prediction of the learning curve shape.

This contribution is thought to be extremely useful to researchers or users that will use the LayoutLM model with similar datasets when planning the annotation process, as the number of samples that need annotation may be confidently predicted at a very early stage of the process.

6- Declarations

6-1-Author Contributions

Conceptualization, F.C. and M.C.; methodology, F.C. and M.C.; software, F.C.; validation, F.C. and M.C.; investigation, F.C. and M.C.; data curation, F.C.; writing—original draft preparation, F.C.; writing—review and editing, M.C.; visualization, F.C.; supervision, M.C.; project administration, M.C.; funding acquisition, M.C. All authors have read and agreed to the published version of the manuscript.

6-2-Data Availability Statement

Data used in this work are available. The FLH dataset can be downloaded at the following link: <https://zenodo.org/records/6371710>

6-3-Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6-4-Institutional Review Board Statement

Not applicable.

6-5-Informed Consent Statement

Not applicable.

6-6-Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

- [1] Peres, F., & Castelli, M. (2021). Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development. *Applied Sciences*, 11(14), 6449. doi:10.3390/app11146449.
- [2] Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295–316. doi:10.1016/j.neucom.2020.07.061.
- [3] Kalayeh, H. M., & Landgrebe, D. A. (1983). Predicting the Required Number of Training Samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(6), 664–667. doi:10.1109/TPAMI.1983.4767459.
- [4] Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., & Zhou, Y. (2017). Deep learning scaling is predictable, empirically. arXiv preprint arXiv:1712.00409. doi:10.48550/arxiv.1712.00409.
- [5] Beleites, C., Neugebauer, U., Bocklitz, T., Krafft, C., & Popp, J. (2013). Sample size planning for classification models. *Analytica Chimica Acta*, 760, 25–33. doi:10.1016/j.aca.2012.11.007.
- [6] Dobbin, K. K., & Simon, R. M. (2007). Sample size planning for developing classifiers using high-dimensional DNA microarray data. *Biostatistics*, 8(1), 101–117. doi:10.1093/biostatistics/kxj036.
- [7] Dobbin, K. K., Zhao, Y., & Simon, R. M. (2008). How large a training set is needed to develop a classifier for microarray data? *Clinical Cancer Research*, 14(1), 108–114. doi:10.1158/1078-0432.CCR-07-0443.
- [8] Kier, C., & Aach, T. (2006). Predicting the benefit of sample size extension in multiclass k-NN classification. 18th International Conference on Pattern Recognition (ICPR'06). doi:10.1109/icpr.2006.942.
- [9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30, 1-11, NeurIPS Proceedings, Long Beach, California, United States.
- [10] Viering, T., & Loog, M. (2023). The Shape of Learning Curves: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6), 7799–7819. doi:10.1109/tpami.2022.3220744.
- [11] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1), 43–76. doi:10.1109/jproc.2020.3004555.
- [12] Frey, L. J., & Fisher, D. H. (1999). Modeling decision tree performance with the power law. *Seventh International Workshop on Artificial Intelligence and Statistics*. PMLR, 3-6 January, 1999, Fort Lauderdale, United States.
- [13] Hess, K. R., & Wei, C. (2010). Learning Curves in Classification with Microarray Data. *Seminars in Oncology*, 37(1), 65–68. doi:10.1053/j.seminoncol.2009.12.002.

- [14] Brumen, B., Rozman, I., Heričko, M., Černezel, A., & Hölbl, M. (2014). Best-fit learning curve model for the C4.5 algorithm. *Informatica (Netherlands)*, 25(3), 385–399. doi:10.15388/Informatica.2014.19.
- [15] Singh, S. (2005). Modeling performance of different classification methods: deviation from the power law. Project Report, Department of Computer Science, Vanderbilt University, Nashville, United States.
- [16] Figueroa, R. L., Zeng-Treitler, Q., Kandula, S., & Ngo, L. H. (2012). Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, 12(1), 1–10. doi:10.1186/1472-6947-12-8.
- [17] Last, M. (2007). Predicting and Optimizing Classifier Utility with the Power Law. Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Nebraska, United States. doi:10.1109/icdmw.2007.31.
- [18] Cortes, C., Jackel, L. D., Solla, S., Vapnik, V., & Denker, J. (1993). Learning curves: Asymptotic values and rate of convergence. *Advances in Neural Information Processing Systems*, 6, NeurIPS Proceedings, Long Beach, California, United States.
- [19] Kolachina, P., Cancedda, N., Dymetman, M., & Venkatapathy, S. (2012, July). Prediction of learning curves in machine translation. Proceedings of the 50th Annual Meeting of the Association for Computational, 8-14 July, 2012, Jeju Island, South Korea.
- [20] Leite, R., & Brazdil, P. (2004). Improving Progressive Sampling via Meta-learning on Learning Curves. *Machine Learning: ECML 2004. Lecture Notes in Computer Science*, 3201, Springer, Berlin, Germany. doi:10.1007/978-3-540-30115-8_25.
- [21] Hoiem, D., Gupta, T., Li, Z., & Shlapentokh-Rothman, M. (2021). Learning curves for analysis of deep networks. International conference on machine learning, 18-24 July, 2021, Virtual Event.
- [22] Mukherjee, S., Tamayo, P., Rogers, S., Rifkin, R., Engle, A., Campbell, C., Golub, T. R., & Mesirov, J. P. (2003). Estimating Dataset Size Requirements for Classifying DNA Microarray Data. *Journal of Computational Biology*, 10(2), 119–142. doi:10.1089/106652703321825928.
- [23] Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021). “Everyone wants to do the model work, not the data work”: Data Cascades in High-Stakes AI. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. doi:10.1145/3411764.3445518.
- [24] Castelli, M., Pinto, D. C., Shuqair, S., Montali, D., & Vanneschi, L. (2022). The Benefits of Automated Machine Learning in Hospitality. *Emerging Science Journal*, 6(6), 1237-1254. doi:10.28991/ESJ-2022-06-06-02.
- [25] ICDAR. (2019). Overview - ICDAR 2019 Robust Reading Challenge on Scanned Receipts OCR and Information Extraction. Robust Reading Competition. Available online: <https://rrc.cvc.uab.es/?ch=13> (accessed on April 2023).
- [26] Cruz, F., & Castelli, M. (2022). Dataset of personal invoices and receipts including annotation of relevant fields. 16 October 2022, Version v1. doi:10.5281/ZENODO.7213544.
- [27] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. doi:10.1145/3394486.3403172.
- [28] Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. *Advances in Neural Information Processing Systems*, NeurIPS Proceedings, 28, 1-9.
- [29] Nakayama, H. (2018). Chakki-works/seqeval: A Python framework for sequence labeling evaluation (named-entity recognition, pos tagging, etc...). GitHub, San Francisco, United States. Available online: <https://github.com/chakki-works/seqeval> (accessed on July 2023).
- [30] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272. doi:10.1038/s41592-019-0686-2.
- [31] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12, 2825-2830.