A Model for Estimating the Economic Costs of Computer Vision Systems That Use Deep Learning

Neil Thompson*[1], Martin Fleming[3], Benny J. Tang[1], Anna M. Pastwa[1, 4], Nicholas Borge[2], Brian C. Goehring[2], Subhro Das[2]

[1]MIT,
[2]IBM
[3]The Productivity Institute, Varicent
[4]University of Warsaw

Abstract

Deep learning, the most important subfield of machine learning and artificial intelligence (AI) over the last decade, is considered one of the fundamental technologies underpinning the Fourth Industrial Revolution. But despite its record-breaking history, deep learning's enormous appetite for compute and data means that sometimes it can be too costly to practically use. In this paper, we connect technical insights from deep learning scaling laws and transfer learning with the economics of IT to propose a framework for estimating the cost of deep learning computer vision systems that achieve a desired level of accuracy. Our tool can be of practical use to AI practitioners in industry or academia to guide investment decisions.

Introduction

As technologies mature, the economic impacts of their implementation can be reasonably anticipated with some level of accuracy, e.g. the cost and benefits of a database. Such assessment is important for effective resource-allocation decisions. But case studies that we conducted with industry about deep learning systems indicate that they are far from reaching this level of predictability. This lack of precision in costs and benefits creates investment uncertainty, because while these systems can reach impressive, nearly human-level performance in tasks such as vision and language, they also come with an enormous appetite for compute and data that can be costly. In the absence of complete information, decision makers are less informed and often invest in alternative projects with a lower expected value. Project uncertainty also likely hinders useful industry implementations of deep learning because of limited assurance that it will be money well spent. Given the rapidly rising economic and environmental costs of deep learning (Thompson et al. 2020; Strubell, Ganesh, and McCallum 2020), this lack of information needed for planning is problematic.

In this paper, we provide evidence that a better way is possible. In particular, we show that regularities in deep learning scaling and transfer learning can be the basis for making predictive estimates for the cost and accuracy of new systems. We operationalize this method by re-analyzing data from Mensink et al. (2021), and then parameterizing it and validating it with data from 14 real industry implementations. While most of the literature typically relies on repeated experiments using publicly available benchmark datasets, our method bridges the gap and provides industry-tested evidence (Klemetti et al. 2023). Overall, we find promising results that not only can such a model be predictive of deep learning costs, but that the inputs needed would be relatively straightforward for users across industry or academia.

*Contact author: neil_t@mit.edu

Challenges in Resource Allocation

A decision on when and how to invest in deep learning must depend on a reasonable cost-benefit analysis in light of the desired model performance. When assessing deep learning suitability, a notable difficulty lies in accurate assessment of implementation costs (IBM 2021). Availability of adequate cost estimates is important for several reasons. First, in business terms project approvals often depend on the magnitude of a reasonably estimated cost. Second, costs need to be tracked and audited. Third, more practically, knowing costs can allow firms to prioritize experiments that can feasibly achieve desired performance. The ability to select the most promising experiments is an important factor when the speed of solution delivery matters.

When building deep learning systems, the costs of achieving a desired level of accuracy may be too high, given the expected benefits. Indeed, high costs are one of the primary reasons organizations say attempted deployments fail (Wiggers 2019). In fact, the largest deep learning experiments require resources that exceed the budgets of all but a handful of institutions (Ahmed and Wahed 2020).

Faced with cost uncertainty, firms with risk-averse cultures or incentive structures may choose not to invest or only invest in incremental options (Hoskisson, Hitt, and Hill 1993). Prior work has shown that this rarely leads to breakthrough innovations and instead results in lower-payoff economic benefits (Carson et al. 2020). Thus, firms forced to make resource allocation under high uncertainty may make worse decisions.

Transfer Learning

Transfer learning (Pan and Yang 2010; Zhuang et al. 2021) can sometimes be used to overcome some of the challenges. The high-level idea behind this technique is a transfer of knowledge from a source domain, which consists of a large dataset with high-quality labels, to a target domain, where data is limited and labels are not always available, with the goal of improving performance in the latter. In contrast to developing a deep learning model from scratch, where training and testing happens within one domain and one task, transfer learning enables extrapolation to solve new problems. Source domains and tasks for the training can be different from the target ones but need to be related to some extent. For example, a model analyzing cars might still be useful for analyzing snowmobiles because many low-level visual features (lines, edges, etc.) are needed for processing both.

Even though transfer learning allows for cost reduction thanks to knowledge reuse, it can still require substantial resources for the fine-tuning stage. Fortunately, a growing body of research investigates the conditions under which transfer learning works well, e.g., selecting the most optimal source dataset, architecture, or pre-training schemes (Agostinelli et al. 2022).

Typically, the exact cost of running deep learning models is proprietary, and thus can only be inferred by an educated guess, or via anecdote (Sharir, Peleg, and Shoham 2020). There is also little to no requirement for reporting the financial cost of developing, training, and running deep learning or fine-tuned models in published research (Schwartz et al. 2020). Thus, when faced with a decision of whether to invest in these models, companies have to choose from one of two poor decisions: Incur the cost of training to discover how much it would cost and how accurate it will be, or do not invest and potentially lose out on a competitive advantage. In effect, many deep learning projects fail due to cost overruns or because cost concerns prevent innovations being pursued.

Our Contribution

We propose an approach to discovering how much a deep learning investment would cost, how much data is needed, and how accurate it would be at a much lower cost than that of actually training the model. We re-analyze data from earlier transfer learning research (Mensink et al. 2021) and use the empirical results (and a novel measure that we propose) to estimate the cost of adapting a pre-trained deep learning vision model to new proprietary data. Finally, we propose a practitioner-oriented tool that can help predict the costs and performance of new computer vision models developed in this way.

While past research often quantified the resource footprints of deep learning models with metrics such as training time and GPU use (Dehghani et al. 2021), our economic model makes it possible to flexibly predict the costs and data requirements of several complex scenarios even by practitioners who might be less technical.

The results of our study and the economic model can be useful to several stakeholders. First, businesses and organizations may use the tool to estimate the feasibility of their deep learning projects and experiments on proprietary data. In this way, they can have a better understanding into how much data is needed to achieve the desired level of accuracy and at what dollar cost. In a similar vein, applied researchers preparing project proposals would be in a better position to evaluate data requirements and justify the cost of their projects to sponsors. Second, platform engineers may use the economic model to better estimate the resources, such as computation and memory, required to feasibly train and deploy a preferred model in production. In all these cases, better knowledge of the cost and accuracy of the deep learning systems can translate into better resource-allocation decisions.

Related Work

Deep Learning Deployment Costs

Prior literature has often considered the various challenges in deploying deep learning (Paleyes, Urma, and Lawrence 2022) associated with typical stages of a machine learning project life cycle: data management, model learning, model verification, and model deployment (Ashmore, Calinescu, and Paterson 2021). Overall, deep learning project deployment requires not only a large amount of data and copious computing power, but also highly skilled personnel, ongoing maintenance and refinement, and careful management.

The economic cost of the training stage, which entails supplying the data to the chosen model architecture to learn patterns, is considered one of the biggest concerns (Malta, Avila, and Borin 2019).

Deep Learning Performance Prediction

To compare feasibility of models, it is essential to account for both performance and costs (Menghani 2021). When two models perform equally well, the preferred model will be the one costing less. At present, cost considerations associated with deep learning model training and inference are typically fragmented, since the research focus has been predominantly placed on achieving the best possible performance, as measured by accuracy or related metrics (Schwartz et al. 2020; Dehghani et al. 2021).

Various footprint metrics have been proposed for the specific elements of model learning or characteristics of the infrastructure, mostly with the goal to optimize them: training time, number of epochs to converge, forward and backward pass latency, number of parameters in model architecture, number of required labels in datasets for training and validation, hardware/cloud compute, number of floating point operations (FLOPs), RAM consumption, and model disc size. Among them, the most important ingredients contributing to the overall deep learning cost were recognized to be the size of the training dataset, the number of model parameters, and the number of FLOPs required to run an instance of a model to achieve desired performance (Sharir, Peleg, and Shoham 2020).

Past research most often quantified the deep learning model footprint with training time, FLOPs, number of parameters and GPU use. Yet (Dehghani et al. 2021) note that the choice of the reported cost indicators resulted in

an incomplete cost picture leading to potentially misleading conclusions about model efficiency, what they term as *'efficiency misnomer'*. For example, a low number of FLOPs may not mean that the model is actually fast since the FLOPs metric does not account for the degree of parallelism or memory access. Similarly, model-size measures with the number of trainable parameters does not mean it is fast to train or fine tune. Models with very few trainable parameters can be very slow in practice, for example in cases when their parameters are shared across computational steps.

Overall, previous studies rarely attempt to quantify simultaneously the most important drivers of model training costs. To the best of our knowledge an exception is Sharir, Peleg, and Shoham (2020) who predict the cost of NLP models, and Malta, Avila, and Borin (2019) who are interested in the most cost-optimal choice of cloud deployment. It continues to be a challenge to find the most appropriate combination of the essential cost-driving elements for a given problem that achieves a desired performance at minimal cost.

Empirical Approach

Our economic model aims to reduce the uncertainty in resource allocation with a data-backed evaluation of the benefits and costs of investing in a deep learning vision system. This roughly translates to a desired level of performance and the estimated financial cost of training a model to achieve that level of performance. Because we want this tool to be usable by those with or without technical knowledge of deep learning, we focus on considering prediction inputs that are accessible, and we aim to minimize the amount of detail that users need to provide to maximize the information they receive out of it.

We take an empirical approach to building our economic model by leveraging both prior research in transfer learning performance and proprietary data from IBM clients. A flow chart of our model can be found in Figure 1. At the core, our economic model consists of a fitted generalized linear model (GLM). Its parameters and outputs are then used as inputs to a cost model, which generates suggestions for the most cost-effective solutions.

Due to challenges in sourcing data, we limit the scope of our work to only computer vision image-based neural tasks. Specifically, we limit it to classification and semantic segmentation tasks.

Inputs

Target Performance. Research by Thompson et al. (2021) and earlier by Hestness et al. (2017) has shown that scaling a model's performance is governed by power laws. As the desired performance increases, the amount of compute grows polynomially. Practically, this means that equal percentage reductions in error require an equal percentage increase in compute. Consequently, the desired target performance is a core cost driver.

Datapoints. Reflecting the real-world limitation that infinite data is impossible to collect, we allow users to specify as an input both the amount of data that is already usable, as well as that which could realistically collected. We upper bound our model's best possible performance based on the amount of data available.

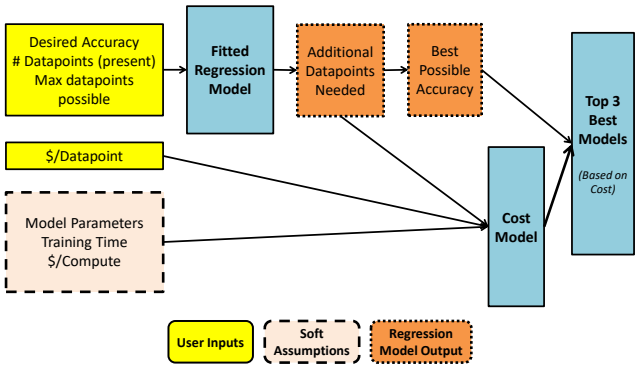

Figure 1: Our economic model architecture, showing user inputs, soft assumptions, intermediate regression model outputs, and final outputs.

In combination with the cost of collecting data, users can weigh the trade-off in marginal cost of acquiring more data versus marginal model performance improvement.

Number of Classes. Intuitively, it takes more data and computation for more challenging tasks. We define task difficulty by the number of classes being classified and compare the relative difficulty of the input task to a very difficult baseline. Optionally, we can obtain even more accurate scores by providing the number of examples in each class. More details on how this is implemented can be found below under *Relative Complexity*.

Soft Assumptions

Common Model Architectures. The size of a given model architecture (in the number of parameters) directly impacts the computational resources required to train it. In discussions with IBM's clients, it became evident that they predominantly engaged in two practices: training a model using existing architectures or fine tuning from a checkpoint of an existing architecture. We did not encounter novel architectures being deployed. Thus, we present a compilation of the most commonly used designs and their model sizes. We include in our list AlexNet (Krizhevsky, Sutskever, and Hinton 2012), EfficientNet-B0 (Tan and Le 2019), EfficientNet-B7 (Tan and Le 2019), Inception-ResNet-v2 (Szegedy et al. 2017), Inception-v3 (Szegedy et al. 2016), Inception-v4 (Szegedy et al. 2017), Mask R-CNN (He et al. 2017), ResNet-32 (He et al. 2016), ResNet-34 (He et al. 2016), ResNet-101 (He et al. 2016), VGG-16 (Simonyan and Zisserman 2015), VGG-19 (Simonyan and Zisserman 2015), YOLOv3 (Redmon and Farhadi 2018), and YOLOv5 (Jocher et al. 2022).

IaaS Cost Estimates. Almost all of the IBM clients chose to use infrastructure-as-a-service with GPUs for training their models, which is a commonly applied deployment paradigm (Klemetti et al. 2023). In combination with model

size and number of datapoints to process, this is a core driver of training cost.

Model testing costs. We assumed that model testing costs are small compared to training costs - consistent with what we observed in our case studies. However, when extensive testing is required (e.g. in a regulated environment), we recommend amending the default number of epochs.

Outputs

Top Three Model Recommendations. Our economic model suggests the three best models based on a lowest-cost prediction. Additionally, it provides information about the best possible performance achievable given the available and collectable data. We choose the lowest cost as it is the most desirable when performance is equal across multiple models. Our model's interface for inputs and outputs can be seen in Figure 2.

Inputs

Task	Classification
Target Accuracy	96.50%
Limit of Datapoints	200000
$ per Datapoints ($)	0.63
Datapoints Number of Classes	
- OR -	
Class Distribution	
Class Label	# Datapoints
Positive	50000
Negative	1000
Total Datapoints	*51000*

Soft Assumptions

Base Model	VGG-16
Pretrain	TRUE
$ per FLOP ($)	3E-11
Epochs to Train	50

Outputs

Best Model	EfficientNet-B0
Max Achievable Accuracy	96.50%
Model Training Cost	$ 9,610.92
Data Cost	$ 93,423.00
Data Needed	149,000
Total Cost	$ 103,033.92

2nd Best Model	Inception-v3
Max Achievable Accuracy	96.50%
Model Training Cost	$ 43,313.31
Data Cost	$ 93,423.00
Data Needed	149,000
Total Cost	$ 136,736.31

3rd Best Model	YOLOv5
Max Achievable Accuracy	96.43%
Model Training Cost	$ 76,317.51
Data Cost	$ 93,423.00
Data Needed	149,000
Total Cost	$ 169,740.51

Figure 2: Model inputs (left) and model outputs (right)

Total Costs. Corresponding to each model, we provide a breakdown of the total costs into model-training cost and additionally collectable data required along with its cost. The calculations for these can be found below.

Regression Model

We implement GLM using the Python `statsmodels` package version 0.12.2 (Seabold and Perktold 2010).

Data. Data used to fit the model come from two sources. The main source of our data is 48 datapoints of semantic segmentation performance from the transfer learning paper by Mensink et al. (2021), which we re-analyze. Usefully, their work also provides us with an "all else equal" analysis on how well different models performed on various tasks with and without the use of transfer learning from a pre-trained checkpoint.

To test the real applicability of this model, we conducted 35 case-study examples of deep learning deployments with IBM clients. Most of these were able to provide broad, qualitative information, but in 14 deployments we have sufficient information to include them in our analysis. We use eight of these to augment our training, and six for testing the predictions of our regression model.

While data from the literature is on semantic segmentation, we also adapt it for use in image classification by conceptualizing it as a complex form of multi-class classification.

Regression Variables. Our approach aims to be generalizable across a range of tasks and datasets. Thus, we define and encode Boolean variables that we intend to use in future work. The core of our regression model's contribution lies in defining concepts of "distance" between two tasks and two datasets, which we call *Relative Complexity* and *Dataset Distance* respectively.

Peformance Error. The model error, either for semantic segmentation calculated as mean $(1 - JaccardIndex)$ averaged over target task classes, or $(1 - Accuracy)$ for classification tasks.

Datapoints. How much data the model was trained on for the target task.

Pretrain. A Boolean variable that indicates whether the model was fine-tuned from a pre-trained model checkpoint. We encode 0 for "no", and 1 for "yes".

Relative Complexity (RC). The relative difficulty of a target task versus a very difficult baseline task, as measured by entropy. This score is calculated by $\frac{TargetTaskEntropy}{BaselineTaskEntropy}$, where a score closer to 0 is an easy target task, and a score closer to 1 is a task that approaches the difficulty of the baseline. The entropy of each task for n classes can be calculated from classic (Shannon 1948) as $H(Y) = -\sum_{k=1}^{n} P(y_k) log P(y_k)$. Given that our baseline upper bounds this score, by default we choose ImageNet-1000 (Russakovsky et al. 2015) for our baseline. However, this is a parameter that can be manually adjusted.

Type. A Boolean variable used in bootstrapping from task performance data. Semantic segmentation is encoded as 1 and classification is encoded as 0.

Dataset Distance. Our measure of "distance" between a target task dataset and a baseline dataset, using sampled embeddings from both. We calculate this by first sampling 10,000 images from each dataset. We then pass these through a pre-trained Inception-V3 model (Szegedy et al. 2016) and average the penultimate fully connected layer's embeddings. We then measure the Wasserstein distance (Kantorovich 1960) between both averaged embedding vectors, producing a no-maximum score where 0 indicates an identical embedding space, and increasing score indicates an increasingly different embedding space.

Regression Equation Setup. We iteratively fit our model and record both statistical significance and R^2. During this process, dataset distance repeatedly failed to meet our statistical significance threshold of 0.05, and was thus dropped.

Our final regression model is parameterized as such:

$$\begin{aligned}\log_{10}(error) = &-0.813 - 0.070 * \log_{10}(datapoints) \\ &- 0.053 * Pretrain * \log_{10}(datapoints) \\ &+ 0.609 * RC + 0.622 * Type\end{aligned} \tag{1}$$

Cost Model

With our fitted GLM, we rearrange the terms in Equation 1 and solve for $datapoints$ with the following equation:

$$data_{needed} = 10^{\frac{\log_{10}(error)+0.813-0.609*RC-0.622*Type}{-0.070-0.053}} \tag{2}$$

Finally, we combine our defined economic model inputs in the following manner to calculate the total cost for a given model.

$$cost_{total} = cost_{training} + cost_{data} \tag{3}$$

$$\begin{aligned}cost_{training} = &\ cost_{perflop} * 2 * modelsize \\ &* epochs * 3 * min(data_{needed}, data_{max})\end{aligned} \tag{4}$$

$$\begin{aligned}cost_{data} = &\ cost_{perdatapoint} \\ &* (min(data_{needed}, data_{max}) - data_{present})\end{aligned} \tag{5}$$

In Equation 3, we break our cost calculation into the cost of training ($cost_{training}$) and the cost of additional data acquisition ($cost_{data}$).

Using a FLOPS-of-compute estimation by Sevilla et al. (2022), we multiply with the per-FLOP cost ($cost_{perflop}$) in Equation 4. We ensure that we take the lesser of the required amount of data and the maximum amount of data that the user can collect.

To calculate the marginal cost of data in Equation 5, we multiply the per-datapoint cost ($cost_{perdatapoint}$) with the difference of the current quantity of available data and the lesser of the required amount of data and the maximum amount of data that the user can collect.

Discussion of Results

Our GLM achieves a highly predictive R^2 of 0.840 with all variables statistically significant at a threshold of 0.05. The corresponding regression details can be found in Table 1. The model's partial regression grid can be found in Figure 3. We verify the predictions of our regression model against our test set. In all but one case, our model was able to approximately predict similarly to our observed error, as seen in Table 2.

We further evaluate the proposed economic model on a number of real-world case studies of deep learning model deployments. In internal discussion with IBM stakeholders and external discussion with IBM clients, we validated our model's usefulness in the closed-problem case where the parameters of the task are well known. In the open-problem case, we identified areas for future improvement.

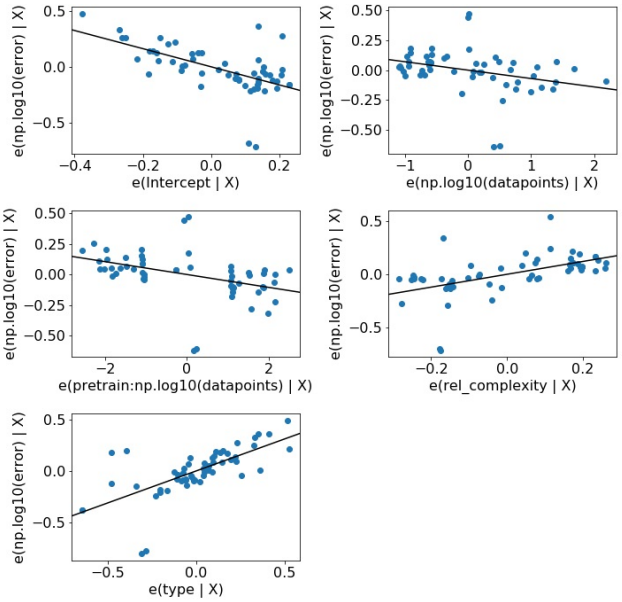

Figure 3: Partial regression grid/added variable plots showing the effect of adding a variable to a model with one or more existing independent variables, with all else held equal.

Where this model shines is when the problem, business case, and inputs are well-defined. Potential users saw the value of our economic model in acceptance testing. One client stated that they "based [their] solution decisions of go/no-go on some soft criteria ... [such as if] there is a feasible opportunity; this provides a go/no-go that's based on data and [is] structural". Other potential users pointed to the usefulness as a resource-allocation tool for prioritization of research and its strength in use cases focused on deployment. In particular, it allows for accurate "amortization of costs over multiple solutions or implementations, such as in grocery stores ... or multiple product [lines]," according to another client.

Other clients offered opportunities for extensions of our work in less defined cases. Different paradigms to approaching training and regularization exists, such as early stopping. A client suggested that being able to compare these paradigms and "knowing convergence early on" would be helpful for them to evaluate not just *what* to train, but also *how* to train. There are also suggested extensions to the cost-benefit analysis, such as incorporating the economic cost of the consequence for errors and the economic benefit of other forms of returns. In the former, the cost of making a mistake classifying mammography scans was given as an example. In the latter, first-mover advantage and positive press were offered as possible returns to be captured. It was also suggested to focus on capturing the nuances of data. For example, how to model stability of data in the real world or to quantify the cost-benefit of remedying dataset class imbalance.

Some clients painted a less rosy picture of the economic model's use. In some cases, clients were more focused on research innovation rather than economic outcome. With other

Variable	Coefficient	Std. Err.	t	$P > \|t\|$	[0.025	0.975]
Intercept	-0.813	0.151	-5.378	0.000	-1.117	-0.510
$\log_{10}(datapoints)$	-0.070	0.029	-2.456	0.018	-0.127	-0.013
Pretrain*$\log_{10}(datapoints)$	-0.053	0.016	-3.367	0.001	-0.084	-0.021
Relative complexity (RC)	0.609	0.136	4.464	0.000	0.335	0.883
Type	0.622	0.093	6.719	0.000	0.436	0.808

Table 1: Results of an OLS regression model predicting $\log_{10}(error)$ with R^2 of 0.840.

Case	Data	RC	Obs. Err.	Pred. Err.
Boston Sci.	50100	0.012	3%	4%
Boston Sci.	61000	0.012	1%	4%
IBM CAO	61000	0.100	1%	5%
Navtech	3000	0.200	6%	8%
Navtech	3000	0.348	12%	9%
Navtech	3000	0.403	45%	1%

Table 2: The inputs and predictions to our regression model, verifying its predictive power against our test set in transfer learning for classification.

clients, their projects were too open ended for this model to be of use, such as not knowing what experiments were needed to be run, or how many, within a single project.

Overall, this feedback suggests that our work makes progress towards a better model for reducing the uncertainty of resource allocation. In fact, these discussions have demonstrated that *so much* uncertainty exists that users wish to expand on the current scope and level of nuance our model provides.

Limitations & Future Work

Our work represents an early attempt to estimate the cost of machine learning deployments ahead of time. However, challenges remain that constrained our scope and that we have identified as opportunities for future work. Many of these challenges relate to limited data availability.

Limited Data. Our primary constraint was only having fourteen datapoints from IBM clients. This constrained us and required bootstrapping on data from prior research. Furthermore, target performance for extremely high or low values at the limits of our training data may produce unreliable predictions due to the extrapolation effect. This also prevented us from accurately capturing the cost of training a model from scratch.

Task. Related to that, our bootstrapped data used semantic segmentation as the task, compared to the IBM client data using classification. While these computer-vision tasks are similar, they are not equivalent to one another, and we expect an impact in our model's predictive capability. Consequently, our definitions of performance also differ - we use mean $(1-JaccardIndex)$ and $(1-Accuracy)$ for semantic segmentation and classification errors respectively.

Domain. The complexities of different domains meant that our economic model only applies to computer vision. We are in the process of exploring how we might apply it to the domains of language, tabular, and time series data.

Other Fixed Costs. Our economic model presented here does not capture the human resource or fixed infrastructure cost required to train these models.

Training Time. We do not capture training time in our model. Specifically, we do not model the impact that different training duration (and regularization strategies) have on model performance. We assume this to be the best model error in all epochs trained with no early stopping for some sufficiently large number of epochs. Running experiments across a variety of tasks and datasets will allow us to model performance as a function of training time used as a model input.

Environmental Cost. While in this paper we focus on the economic cost, we are currently extending the analysis to incorporate the carbon footprint, following (Strubell, Ganesh, and McCallum 2020; Thompson et al. 2020; Dodge et al. 2022). We believe that as models grow larger, it is more responsible to expand the definition of cost from the purely economic to include environmental cost.

Deployment Path. Our deployment path comprises several stages, with initial steps already completed and upcoming ones planned. In late 2022, we developed a user-friendly model prototype in Excel. In February 2023, we hosted a workshop with IBM clients, providing insights into the tool's applications. Post-workshop, we shared the model framework and encouraged alpha/beta testing. Currently, we are gathering feedback and training additional models for validation. By early 2024, we aim to release an updated tool on the IBM intranet, along with the publication of shareable data and code. We also envision future extensions to the model, especially to consider more recently published foundation models.

Conclusion

In this paper, we propose an economic model to estimate the feasibility and cost of achieving desired performance of a pre-trained deep learning model for vision problems. Our method can also provide information on the data needs to achieve particular levels of performance. Various types of business and academic stakeholders can benefit from using the tool to decide on the parameters of their deep learning projects.

Acknowledgements

This research is supported in part by IBM grant. We thank the staff at IBM Research and client companies for providing case studies and feedback that informed this model.

References

Agostinelli, A.; Uijlings, J.; Mensink, T.; and Ferrari, V. 2022. Transferability Metrics for Selecting Source Model Ensembles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7936–7946.

Ahmed, N.; and Wahed, M. 2020. The De-democratization of AI: Deep Learning and the Compute Divide in Artificial Intelligence Research. arXiv:2010.15581.

Ashmore, R.; Calinescu, R.; and Paterson, C. 2021. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. *ACM Computing Surveys*, 54(5).

Carson, R.; Graff Zivin, J. S.; Louviere, J.; Sadoff, S.; and Shrader, J., Jeffrey G. 2020. The Risk of Caution: Evidence from an R&D Experiment. Working Paper 26847, NBER.

Dehghani, M.; Arnab, A.; Beyer, L.; Vaswani, A.; and Tay, Y. 2021. The efficiency misnomer. arXiv:2110.12894.

Dodge, J.; Prewitt, T.; Tachet des Combes, R.; Odmark, E.; Schwartz, R.; Strubell, E.; Luccioni, A. S.; Smith, N. A.; DeCario, N.; and Buchanan, W. 2022. Measuring the carbon intensity of ai in cloud instances. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 1877–1894.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hestness, J.; Narang, S.; Ardalani, N.; Diamos, G.; Jun, H.; Kianinejad, H.; Patwary, M. M. A.; Yang, Y.; and Zhou, Y. 2017. Deep Learning Scaling is Predictable, Empirically. arXiv:1712.00409.

Hoskisson, R. E.; Hitt, M. A.; and Hill, C. W. L. 1993. Managerial Incentives and Investment in R&D in Large Multiproduct Firms. *Organization Science*, 4(2): 325–341.

IBM. 2021. Enterprise AI capabilities. Unpublished data.

Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012; Kwon, Y.; TaoXie; Fang, J.; imyhxy; Michael, K.; Lorna; V, A.; Montes, D.; Nadar, J.; Laughing; tkianai; yxNONG; Skalski, P.; Wang, Z.; Hogan, A.; Fati, C.; Mammana, L.; AlexWang1900; Patel, D.; Yiwei, D.; You, F.; Hajek, J.; Diaconu, L.; and Minh, M. T. 2022. ultralytics/yolov5: v6.1.

Kantorovich, L. V. 1960. Mathematical Methods of Organizing and Planning Production. *Management Science*, 6(4): 366–422.

Klemetti, A.; Raatikainen, M.; Myllyaho, L.; Mikkonen, T.; and Nurminen, J. K. 2023. Systematic Literature Review on Cost-efficient Deep Learning. *IEEE Access*, 11: 90158–90180.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, 1097–1105. Red Hook, NY, USA: Curran Associates Inc.

Malta, E. M.; Avila, S.; and Borin, E. 2019. Exploring the Cost-Benefit of AWS EC2 GPU Instances for Deep Learning Applications. In *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, UCC'19, 21–29. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368940.

Menghani, G. 2021. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys*, 55: 1–37.

Mensink, T.; Uijlings, J. R. R.; Kuznetsova, A.; Gygli, M.; and Ferrari, V. 2021. Factors of Influence for Transfer Learning Across Diverse Appearance Domains and Task Types. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44: 9298–9314.

Paleyes, A.; Urma, R.-G.; and Lawrence, N. D. 2022. Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Computing Surveys*, 55(6).

Pan, S. J.; and Yang, Q. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22: 1345–1359.

Redmon, J.; and Farhadi, A. 2018. YOLOv3: An Incremental Improvement. arXiv:1804.02767.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

Schwartz, R.; Dodge, J.; Smith, N. A.; and Etzioni, O. 2020. Green AI. *Communications of the ACM*, 63(12): 54–63.

Seabold, S.; and Perktold, J. 2010. Statsmodels: Econometric and statistical modeling with Python. In *Proceedings of the 9th Python in Science Conference*, 92–96. Austin, TX.

Sevilla, J.; Heim, L.; Hobbhahn, M.; Besiroglu, T.; Ho, A.; and Villalobos, P. 2022. Estimating Training Compute of Deep Learning Models. https://epochai.org/blog/estimating-training-compute. Accessed: 2023-01-26.

Shannon, C. E. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27: 379–423.

Sharir, O.; Peleg, B.; and Shoham, Y. 2020. The Cost of Training NLP Models: A Concise Overview. arXiv:2004.08900.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

Strubell, E.; Ganesh, A.; and McCallum, A. 2020. Energy and policy considerations for modern deep learning research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09): 13693–13696.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI'17, 4278–4284. AAAI Press.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826. IEEE Computer Society.

Tan, M.; and Le, Q. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 6105–6114.

Thompson, N. C.; Greenewald, K.; Lee, K.; and Manso, G. F. 2020. The computational limits of deep learning. arXiv:2007.05558.

Thompson, N. C.; Greenewald, K.; Lee, K.; and Manso, G. F. 2021. Deep Learning's Diminishing Returns: The Cost of Improvement is Becoming Unsustainable. *IEEE Spectrum*, 58(10): 50–55.

Wiggers, K. 2019. IDC: For 1 in 4 companies, half of all AI projects fail. Accessed: 2023-01-26.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2021. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1): 43–76.