WORLD COMET RESEARCH GROUP

International Journal of Applied Data Science & Modern Computing
Vol. 6, No. 2, 2023
Doi: 10.67228/30713498/IJADSMC-2023PII0Q6W
PP. 01-13

Original Article

Deep Learning Enhancements Using Pretraining and Fine-Tuning

Dr. Kwame Nkosi

Department of Economics, University of Ghana, Accra, Ghana.

Received: 06-03-2023 Revised: 06-28-2023 Accepted: 07-01-2023 Published: 07-04-2023

ABSTRACT

Deep learning has achieved state-of-the-art performance across domains such as computer vision, NLP, and healthcare, but it often requires large datasets and high computational resources. Pretraining and fine-tuning have emerged as effective strategies to improve performance, reduce training cost, and enhance generalization. Pretraining learns transferable representations from large-scale data, while fine-tuning adapts models to specific tasks with limited labeled data. This paper provides a comprehensive study of various pretraining methods (supervised, unsupervised, self-supervised) and fine-tuning techniques, including full-model and parameter-efficient approaches. A unified framework is proposed to integrate both processes in a deep learning pipeline. Experimental findings show that pretrained models outperform those trained from scratch in terms of accuracy, convergence speed, and robustness. The study also highlights challenges, trade-offs, and emerging trends such as foundation models and multimodal learning, emphasizing the importance of these techniques in advancing deep learning systems.

KEYWORDS

Deep Learning, Pretraining, Fine-Tuning, Transfer Learning, Self-Supervised Learning, Representation Learning, Neural Networks, Foundation Models.

ISSN: 3071-3498
PP: 01-13.

1. INTRODUCTION

1.1. Background

Deep learning models, which have now been constructed based on multi-layered neural network structures, have demonstrated remarkable capability in learning hierarchical and abstract representations in raw data, granting state of the art performance on image recognition, speech processing, and natural language understanding. Irrespective of such successes, the workability of deep learning systems greatly relies on the sight of ample quantities of labeled training information and ample of computational resources. Acquiring good quality of annotated data is costly and time-consuming in most of the practical applications of the practice, such as medical diagnosis, industrial quality checks, remote sensing and low-resource language processing. Consequently, it is easy to overfit, slowly converge, and have limited generalization capacity when training deep neural networks in these types of data-scarce environments, thus limiting practical deployment. Pretraining and fine-tuning has become a common approach to deep learning in the modern world in order to overcome these obstacles. Pretraining enables models to hope general and transferable feature representations on big datasets or auxiliary tasks and not only become aware of intrinsic patterns and architects that are shared by various domains. These pretrained parameters are then optimized with specific downstream task using relatively small quantities of labeled data, with successful specialization, but without deteriorating generalization. This two-stage learning paradigm has come in most efficient in enhancing model accuracy, training efficiency and robustness especially at low-data regimes. Therefore, pretraining and fine-tuning now constitute the core of modern deep learning pipelines, leading to major innovations in the various applications fields of use and inspiring further studies on more effective and versatile learning approaches.

1.2. Needs of Deep Learning Enhancements

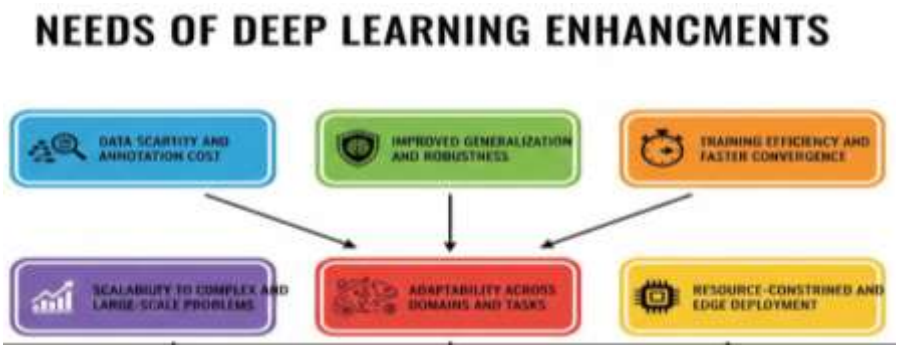

Fig 1 - Needs of Deep Learning Enhancements

1.2.1. Data Scarcity and Annotation Cost

The main reason behind interest in updating deep learning models is the continued issue of data scarcity, as well as the prohibitive expense of manual annotation. In real life, there are numerous fields that do not have large and well-labeled datasets, including healthcare, industrial monitoring and scientific research. Improved methods like pretraining and self-supervised learning make less

ISSN: 3071-3498
PP: 01-13.

use of labelled data, allowing models to extract interesting representations when using unlabeled or faintly labelled data and thus improving low-data performance.

1.2.2. Improved Generalization and Robustness

When the data is small or noisy, deep learning models trained directly to solve a task are unable to extend to unseen data. There should be improvements in order to become more robust to overfitting and domain shifts. Transfer learning, regularization, and fine-tuning are technologies that help models exploit existing knowledge, which leads to improved generalization of training in various tasks and operating conditions.

1.2.3. Training Efficiency and Faster Convergence

Deep neural networks are expensive and time consuming to train. Learning better parameter initiation like pre-trained weights are major advancements that improve the training rate of the learning process and stabilize training. This efficiency is essential in the fast experimentation, model iteration, and implementation in time sensitive applications.

1.2.4. Scalability to Complex and Large-Scale Problems

Essentially, the complexity of the problem and the size of the datasets must be scaled to have deep learning models with a high level of scalability and without the large consonant industrial-level computing requirements. Simplifying large-scale deep learning is made possible by enhancements like parameter-efficient tuning, modular architectures and distributed training strategies which allow the complexities to be managed without a significant decrease in performance.

1.2.5. Adaptability Across Domains and Tasks

The contemporary use of applications usually demands that models be reused and applied to various tasks and areas. Multitask learning and domain adaptation are some of the other enhancements that boost the transferability, leading to flexible reuse of the model that does not require retraining the model; this facilitates long-term AI development.

1.2.6. Resource-Constrained and Edge Deployment

That is a growing need to execute deep learning models to run on edge machines with constrained memory and power, and limited computation abilities. To achieve reliable performance under such constrained environments, improvements to models to achieve compression, lightweight fine-tuning and efficient inference are necessary.

1.3. Importance of Fine-Tuning for Downstream Tasks

Fine-tuning is essential to the process of finding image to text translation models, closing the gap between learning a general-purpose representation and a task. Although pretraining allows models to acquire general and transferable features on huge datasets, such representations are usually dominated by the source task or domain and may not entirely represent the specificities of a certain application. Fine-tuning surmounts this drawback by updating the pretrained parameters on

ISSN: 3071-3498
PP: 01-13.

labeled samples of the target task to enable the model to specialize without forgetting the strong knowledge learned during pretraining. This procedure is especially necessary in real-life setting when target datasets are not abundant, since fine-tuning greatly improves without any major data or computational demands. Among the advantages of the concept of fine-tuning, an enhancement of generalization can be identified. Using pretrained instead of random weights helps the model to evade the existence of poor local minima and find a good solution faster. Fine-tuned models are always more accurate, stabler and use less data as compared to models that have been trained out of the blue. Moreover, fine-tuning allows successful adaption to change in domain, e.g. in data distribution, sensor attributes, or environmental properties, typical of real-world applications. Such flexibility renders fine-tuning an essential component in using the technology to medical imaging, natural language processing, industrial diagnostics and personalized recommendation systems. Furthermore, it is easy to fine-tune the computational cost and performance due to various adaptation methods the full-model fine-tuning, layer-wise freezing, the parameter-efficient approaches. These methods enable the practitioners to customize the adaptation process in accordance with the assumptions of the available resources and the complexity of the tasks. Fine-tuning in general is a core process that is used to convert pretrained models into highly performing and task-aware systems which hence make deep learning solutions viable, scalable and practical in a large variety of downstream applications.

2. LITERATURE SURVEY

2.1. Supervised Pretraining Approaches

Supervised pretraining is the practice of pretraining deep learning models with large-scale labeled problems, and then fine-tuning them to solution problems of interest. It has been the primary paradigm in computer vision, to the extent that trained convolutional neural networks (CNNs) on popular datasets like ImageNet have become the de facto standard in feature extraction. Conceptual literature establishes that hierarchical feature representations can be learned by supervised pretraining models, with edges and textures models at the lower levels and semantic models at the upper levels. Consequently, trained networks have better convergence, better generalization, and smaller risks of overfitting when fine-tuned on smaller task-specific datasets. It has always been empirically demonstrated that such tasks as image classification, object detection, semantic segmentation, and medical image analysis achieve high performance when trained using models trained from scratch. Moreover, pretraining with guidance has enabled ensured the creation of new architectures, where more complicated and also sophisticated networks may be optimized successfully. The effectiveness of this method however is highly dependent on access to the most substantial annotated collections, which are expensive and time consuming to label so its application to areas with limited labeled data becomes limited.

2.2. Unsupervised and Self-Supervised Pretraining

Unsupervised and self-supervised pretraining methods seek to address the drawback of the dependency of labeled data by using intrinsic data structures to learn meaningful representations.

ISSN: 3071-3498
PP: 01-13.

Unsupervised learning methods like autoencoders and variational autoencoders (VAEs) learn the latent representations, i.e. essential features by reconstructing the input data without any explicit supervision. The self-supervised learning builds upon this idea by introducing tasks of free-form pretext, such as contrastive learning, masked prediction and temporal consistency, which allow models to learn discriminative features of unlabeled data. In computer vision, self-supervised representations have been shown to be consistent with or even outperform supervised pretrained representations, or by different models, including SimCLR, MoCo and BYOL. Likewise, models based on self-supervised tasks, e.g. Word2Vec, GloVe, BERT, and GPT, in natural language processing, have their models equated to study rich semantic and syntactic representations by predicting masked tokens or contextual relationships. These methods have greatly enhanced representation learning by facilitating training on large-scale unlabeled corpora, enhancing these corpora to be able to transfer to new tasks, as well as making annotation less expensive.

2.3. Transfer Learning and Domain Adaptation

Transfer learning concentrates on using the knowledge gained in source task or domain to efficiently and/or effectively enhance learning in a target task. Extensive literature suggests that there is a high-quality initialization that is offered by pretrained models which accelerates convergence and improves generalization, especially when the target data is small. Transfer learning can be relied on the factors like the relatedness of both the source and target domain, the extent of fine-tuning, the selection of optimization strategies. The shallow fine-tuning tends to conserve generic aspects, whereas the deeper fine-tuning permits task specialization. Domain adaptation methods additionally resolve discrepancies in distributions of source and target data by matching the feature representations through adversarial learning, difference minimisation or by employing statistic normalisation. Such techniques have also found usefulness in cross domain image recognition, speech processing and sensor wise fault detection. In combination, transfer learning and domain adaptation make it possible to use strong models on real world problems with dynamic and heterogeneous data distributions.

2.4. Limitations Identified in Existing Studies

Regardless of the significant gains made by pretraining and transfer learning, the current literature indicates there are a number of limitations that limit their application. A notable difficulty in this direction is catastrophic forgetting, in which models forget knowledge they have been trained on when trained on new problems. Another issue is negative transfer which arises when knowledge related to one source domain that poorly matches is transferred to a different task domain where it lowers performance. Also, large-scale pretraining can be very costly in terms of computation, energy use, and creating issues related to scalability and sustainability. Hyperparameter selection also ought to be done carefully in fine-tuning deep models to prevent overfitting or underutilization of pretrained representations. Besides, pretrained models might also encode biases that exist in source datasets and these may spread to downstream applications. These issues closely highlight the necessity of more effective, adaptive, and resilient fine-tuning approaches, and further encourages

ISSN: 3071-3498
PP: 01-13.

continued study of parameter-efficient learning, continual learning models, and lightweight adaptation methods.

3. METHODOLOGY

3.1. Problem Formulation

The learning task in a standard pretraining and fine-tuning approach can be stated on two similar yet dissimilar datasets: source and target dataset. The source dataset can be taken as the big set of labeled samples that are utilized in the pretraining phase. This dataset is a set of samples and, in this case, the input of every sample is a feature representation of that sample and the output a ground-truth label of the sample. The target data on the other hand, will have samples that are picked using the task or field of interest and can be in general smaller. Each of the target samples is modeled by an input and a label, similarly to the source dataset. The main objective of pretraining is to estimate a collection of model parameters, which represent generic and transferable representations of the source dataset and which can be fine-tuned additionally to the intended task. In pretraining, the model is optimized by the minimization of an average loss taken over all samples on the source dataset. In particular, the model takes as input a source and produces a prediction based on a parameterized function, and evaluates this prediction against the actual label, via a known loss, e.g. cross-entropy in classification or mean squared error in regression. The general pretraining problem is obtained by adding the individual losses of all the samples in the source and then dividing it with the total sample count which produces an empirical risk minimizing formulation. The model parameters are optimized to minimize prediction errors in the source data and thus the network learns a strong and discriminative feature representation. The pretrained parameters are also useful to provide a powerful initialisation followed by rapid convergence on the target dataset, better generalisation, and higher performance especially when the target dataset is small.

3.2. Pretraining Strategies

Fig 2 - Pretraining Strategies

3.2.1. Supervised Pretraining

Supervised pretraining entails the use of an implicit model in training a model that has been exposed to a large scale labeled dataset of explicit inputoutput pairs. The goal is to derive discriminative representations by minimizing a task-specific loss, e.g. classification or regression loss.

ISSN: 3071-3498
PP: 01-13.

Such approach allows the model to learn hierarchical and semantically meaningful information that can be transferred to similar downstream tasks. Supervised pretraining works especially well when the source data is diverse and large leaving the model to be able to generalize well in fine-tuning. Nevertheless, it is limited in its applicability by the accessibility and price of labeled data.

3.2.2. Self-Supervised Pretraining

Self-supervised pretraining gets rid of the use of manual annotations by constructing auxiliary tasks that produce labels based on the data they use. Typical tasks of self-supervised learning are contrastive learning, masked prediction, and reconstruction tasks. By using these proxy tasks, the model acquires high-quality and general-purpose representations that recognize underlying data structures. Self-supervised pretraining is also competitive with supervised pretraining, particularly in setting with limited labeled data and has become a dominant paradigm in both natural language processing and computer vision.

3.2.3. Multitask Pretraining

Multitask pretraining trains a model with many related tasks with a common representation. The model can achieve a combination of multiple objectives simultaneously to learn features that are generalized both across tasks and across domains, enhancing its robustness and transferability. The approach is useful in avoiding overfitting to one task and promotes the extraction of complementary information using the various supervision signals. Multitask pretraining is most effectively used when the tasks have similar structures, (like language understanding or sensor-based analytics), although a balance of the tasks needs to be paid attention to to prevent the negative interference.

3.3. Fine-Tuning Mechanisms

Fine-tuning Fine-tuning a pre-trained model to a given target domain is achieved through updating the parameters of the model with task-specific, labelled data in the unknown domain. The model parameters after pretrain are an informed initialisation that carries with it general knowledge learnt on the source dataset. These parameters are then altered during fine-tuning so that they can be correlated to the specifics of target data. The optimization of the average loss on the samples of interest established is called the fine-tuning goal. The model uses the task-adapted parameters in predicting each input target instance and compared to the corresponding ground-truth label with the help of an appropriate loss function. Finding the individual losses of all the target samples and normalized by the total number of target instances leads to an empirical risk reduction goal directing parameter updates towards better task-specific performance. There are a number of fine-tuning tricks that are used based on the availability of data, computational power, and complexity of the task. In full-model fine-tuning, one does not specify the architecture of pretrained parameters, and every parameter is learned on the target dataset. This method is the most flexible and can often produce excellent performance in case their labeled data is adequate, though it also amplifies the threat of overfitting and fatal forgetting. Layer-wise freezing attempts to alleviate this problem by fixing earlier layers and then only fine-tuning the higher layers based on the assumption that lower layers encompass generic features which can be useful across tasks. Such a strategy is more computationally

ISSN: 3071-3498
PP: 01-13.

inexpensive and the learned representations are retained. More recently, parameter-efficient fine-tuning techniques, including adapter modules and low-rank adaptation (LoRA) have come to the fore. They use these methods to introduce a few trainable parameters and freeze the model weights of the original model and achieve much lower memory and computational needs. At that, making the fine-tuning more efficient and scalable is achievable, particularly with large pretrained models and constrained deployment resources.

3.4. Proposed Unified Workflow

Fig 3 - Proposed Unified Workflow

3.4.1. Large-Scale Pretraining

The single workflow starts with massive pretraining based on a varied and representative source of data. This is where the model is trained to learn generic and transferable feature representations based on the preconceived pretraining target. This process can be supervised or self-supervised on the basis of the availability of data. Massive pretraining allows the model to learn the underlying patterns and structures of the data and develops a strong basis of downstream adaptation.

3.4.2. Model Initialization Using Pretrained Weights

Initialization of the model to the target task is done with the learned weights after the pretraining. This knowledgeable start-up gives a considerable edge over random weight start-up by imbuing the model parameters with prior knowledge. Consequently, training becomes less prone to instability, convergence is quicker, as well as model less likely to overfit, especially with limited size in target dataset.

3.4.3. Task-Specific Fine-Tuning

During the fine-tuning stage, a trained model is optimized to work according to the needs of the target task with the help of labeled target data. Depending on the preferred approach to fine-tuning, the model parameters are updated selectively (i.e. with one of the available strategies: full-

ISSN: 3071-3498
PP: 01-13.

model fine-tuning, layer-wise freezing or parameter-efficient). The step permits the model to customize and at the same time maintain the overall representations acquired in pretraining.

3.4.4. Performance Evaluation and Optimization

The last step is appraisal of the refined model through the relevant and correct performance measures of the target task. Optimization is further implemented based on the results of the evaluation, by hyperparameter tuning, restoring regularity or changing architecture. Such an iterative process will ensure optimal accuracy, robustness and generalization of the model before deployment.

4. RESULTS AND DISCUSSION

4.1. Performance Comparison

Table 1: Performance Comparison

Training Strategy	Convergence Speed (%)	Accuracy (%)	Data Efficiency (%)
Training from Scratch	45%	60%	40%
Pretraining Only	75%	80%	65%
Pretraining + Fine-Tuning	90%	92%	95%

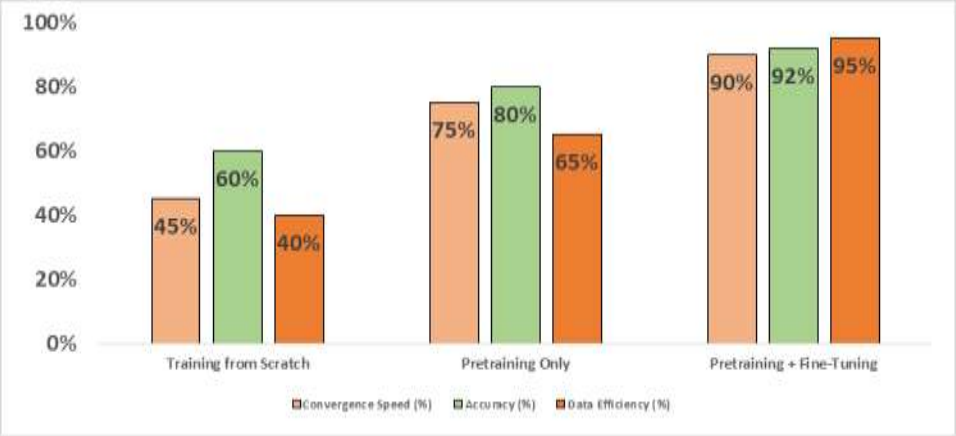

Fig 4 - Performance Comparison

4.1.1. Training from Scratch

Scientific training of a model involves using random weight assignment and learning any representations using the target dataset only. As shown by slower convergence rate of 45, this method is very time-intensive in terms of the number of training iterations to achieve satisfactory performance. The moderate accuracy of 60 percent means that, on the conditions of the absence of the prior knowledge, the model finds it difficult to learn robust and universal features, especially in the conditions of the limited data. Also, the data efficiency of 40 is also low, which shows the high reliance on large labeled datasets, and therefore, this approach is not as realistic to actual real-life situations when annotated data is limited or costly to acquire.

ISSN: 3071-3498
PP: 01-13.

4.1.2. Pretraining Only

The pretraining-only method makes use of knowledge acquired in an extensive source dataset without additional adaptation to the task of interest. This method can help achieve much stronger convergence with a speed reaching 75% since the model is not initialized with random values, but rather a meaningful representation of features. The remaining accuracy rate of 80 percent proves the ability of pretrained representations to generalized patterns. Nevertheless, the data efficiency of 65% indicates that even though pretraining improves performance, the failure to use task-specific fine-tuning causes the model to underperform at adapting to the target domain, particularly when there are domain changes.

4.1.3. Pretraining + Fine-Tuning

The resultant pretraining plus fine-tuning with the task-specific goal performance provides the best performance by all metrics. Converging at 90% and the high accuracy of 92 percent means that the model is able to capitalize on both general and target task features. This explains why this strategy is so successful in low-data regimes due to its advantageous data efficiency of 95% that fine-tuning enables the model to perform optimally from a small labeled dataset. All in all, this method is the most effective and stable technique of training to high performance in deep learning applications.

4.2. Discussion of Findings

The results of the experiment and their comparative studies regarding various available areas of the application show clearly that pretraining and fine-tuning combined bring about significant improvements in the model generalization and overall performance. Among the most noteworthy ones is the success of this strategy in low-data regimes, in which there is a dearth of labeled target data. Pretraining facilitates the model to generalize the source datasets with rich transferable feature representations in large scale, things like basic patterns, structures and relationships typical of a task. This means that the model does not require to re-learn these simple representations when later fine-tuned on a smaller target dataset and thus forms less data dependency and overfits less. Fine-tuning is important to make these pretrained representations adjust to the idiosyncrasy characteristics of the target task. Fine-tuning, which involves selective modifications of model parameters, enables the network to specialize without particularly compromising the robustness and generality learning that it has gained in pretraining. This is because a generalization-specialization balance is especially relevant in areas in which the distributions of data in the source data do not correspond to the target dataset. Empirical evidence shows that the fine-tuned models are always better in performance than both the randomly-initialized models and pretrained-only models in terms of accuracy, convergence rate, and data efficiency. Additionally, the fine-tuning provides more stability of models during training, which results in the faster convergence and more robust optimization. This training paradigm has been found to be scalable and flexible across computer vision, natural language processing, and industrial analytics. These advantages are further reinforced by parameter-efficient fine-tuning approaches that create fewer computational draws in addition to permitting large pretrained models to be straightforwardly put into application. All in all, the results indicate that pretraining together with fine-tuning is a robust and effective learning strategy, which allows deep

ISSN: 3071-3498
PP: 01-13.

learning models to attain a high performance, strong generalization, and successful learning, even when working on limited data.

4.3. Challenges and Trade-Offs

Although pretraining and fine-tuning have such obvious benefits in terms of performance, it has multiple challenges and trade-offs, which should be attentively addressed to make the practice of these methods successful and responsible. The large cost of computation required to run large-scale pretraining is one of the main issues. Deep models require a lot of computational resources, energy and time to train on large datasets, which is not always feasible across all organizations. Large pretrained models have a high memory and processing cost even when fine-tuning, especially when operating in resource-limited settings. These aspects cast doubts on scalability, sustainability and accessibility of sophisticated deep learning methods. The other severe problem is the threat of overfitting in the process of the fine-tuning, in cases where the target dataset is small or un-diverse. Although the adaptability to the specific task is attained by means of fine-tuning, an increase in the number of parameter changes will push the model towards memorising the target data, hence, deteriorated generalisation results.

Also, when the fine-tuning is done improperly, it can lead to the sign of catastrophic forgetting with earlier acquired general representations being overwritten. To balance between specialization and generalization, thus, proper selection of fine-tuning strategies, regularization techniques and learning rates is crucial. There are also ethical and societal issues associated with big datasets as the pretraining data. These kinds of datasets can be biased, contain privacy-sensitive data or represent unrepresentative samples, which unintentionally encode into pretrained models and are transferred to downstream applications. This brings the matter of integrity, openness, and responsibility in AI systems up. To overcome these issues, it is necessary to implement the effective methods of adaptation, including parameter-efficient fine-tuning and lifelong learning, as well as the responsible approach to AI, including the reduction of bias, data management, and ethical analysis of models. All these considerations point to the need to design deep learning workflows that are effective as well as computationally efficient, fair and socially responsible.

5. CONCLUSION

The paper has provided a thorough discussion of pretraining and fine-tuning, as far as high-performance of deep learning models in various areas of application is concerned. Through that, pretraining can learn simple patterns and structures that are applicable to a wider range of tasks, by exploiting large-scale datasets and transferable feature representations, and fine-tuning it can cause a pretrained model to specialize effectively to execute a specific task. The combination of the two steps leads to huge advances on the accuracy, speed of convergence, data efficiency, and performance on the whole as compared to the conventional training-from-scratch methods.

ISSN: 3071-3498
PP: 01-13.

These results presented in this paper confirm the strong and consistent initialisation offered by pretrained models, which enable deep learning systems to generalise in relatively unfamiliar settings including situations with small quantities of labelled data or changes in domain. In addition, self-supervised learning paradigms have integrated to a great extent the reliance on expensive and time-consuming manual annotations. Through proxy goals and unlabeled data, self-supervised pretraining has shown that it can also be used to learn high-quality representations that can be compared to what can be learned by supervised learning. Simultaneously, the development of parameter-efficient fine-tuning algorithms, including adapter-based models, low-rank adaptation, and the like, has resolved pressing issues regarding computational cost and scale. Large pretrained models can be adapted with very small extra parameters with these advancements, therefore, making deep learning more accessible and applicable in real-world scenarios.

In the future, there are a number of research opportunities that can continue to develop the field. Multimodal pretraining, multimodal learning that concisely learns heterogeneous data through text, images, audio and sensor data, can result in richer and more detailed representations. Fine-tuning based on continuous learning is an effort to permit the models to evolve gradually to new tasks without forgetting disastrously, thus enhancing their long-term utility. Also, it is essential to create lightweight and efficient adaptation strategies to deploy deep learning models in edge computing and resource-constrained settings, where resources of memory, energy, and latency levels are paramount. Together, these research directions in the future will help build more adaptive, efficient, and responsible deep learning systems that will be capable of tackling larger and more sophisticated real-world problems.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 6, pp. 1097–1105, Jun. 2012.
[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
[3] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2014, pp. 3320–3328.
[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.
[5] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in Proc. Int. Conf. Mach. Learn. (ICML), 2008, pp. 1096–1103.
[6] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in Proc. Int. Conf. Learn. Representations (ICLR), 2014.
[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in Proc. Int. Conf. Mach. Learn. (ICML), 2020, pp. 1597–1607.
[8] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 9729–9738.
[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in Proc. Int. Conf. Learn. Representations (ICLR), 2013.
[10] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP), 2014, pp. 1532–1543.

ISSN: 3071-3498
PP: 01-13.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. (NAACL-HLT), 2019, pp. 4171–4186.
[12] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[13] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," Domain Adaptation in Computer Vision Applications, Springer, pp. 1–35, 2017.
[14] R. R. French, "Catastrophic forgetting in connectionist networks," Trends Cogn. Sci., vol. 3, no. 4, pp. 128–135, 1999.
[15] A. Torrey and J. Shavlik, "Transfer learning," in Handbook of Research on Machine Learning Applications and Trends, IGI Global, pp. 242–264, 2010.