Accelerating Machine Learning Research Using Transfer Learning

** R. Rajkumar*
*** Arnav Kaushal*
**** Aishik Saha*

Abstract

Research and development in machine learning involves a continuous cycle of training, testing, and tuning. This is generally a time and computation intensive process, and therefore, leads to a slow experiment cycle. We intended to accelerate this cycle by using the Transfer Learning technique. Using transfer learning we can produce the same results in a matter of minutes, which previously took weeks to generate. This technique is also highly beneficial when the training data is scarce and computing infrastructure is limited. Transfer learning has been applied to many fields ranging from image classification, medical diagnosis of tumors, Diabetic Retinopathy, generating captions, self-driving cars etc. Transfer learning is considered to be the next driver of machine learning success. In this paper, we introduce transfer learning, describe its implementation, and talk about its various applications and benefits.

Keywords : Accelerate, machine learning, research, transfer learning

I. INTRODUCTION

Transfer Learning also called **Inductive Transfer** is a machine learning technique that uses the stored knowledge acquired while solving one problem for solving a different but related problem. For example, knowledge acquired while learning to recognize typewriters can be used to recognize keyboards. In this paper, we focused on applying transfer learning to convolutional neural networks. This technique will allow for the acceleration of the machine learning research cycle [1] by making the training process less time and computation intensive [3]. This is due to the search space getting reduced because of existing knowledge. This also helps when the training data is scarce or cannot be easily obtained. For example, acquiring medical datasets is very difficult and these are also not available in abundance. Thus, the reuse of available models to solve similar problems is beneficial considering time and computation needed. Transfer learning coupled with the latest technologies in machine learning can accelerate the research cycle and lead to more findings by researchers. Specialized hardware like **Graphic Processing Units (GPUs)** and the new **Tensor Processing Units (TPUs)** can further enhance acceleration and lead to a future where artificial intelligence is democratized and available for use by software engineers and researchers alike.

II. TRANSFER LEARNING

A. Convolutional Neural Network (CNN)

In machine learning, a convolutional neural network also referred to as CNN, or ConvNet is a class of deep, feed-forward artificial neural network. CNNs have successfully been applied to analyzing visual imagery. Models with a convolutional neural network architecture solely win ImageNet [2] and similar challenges. These produce the best accuracy till date for recognizing objects and in some cases achieve better recognition than humans. Biological processes in which the connectivity pattern between neurons is inspired by the organization of the animal visual cortex inspired convolutional networks. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence

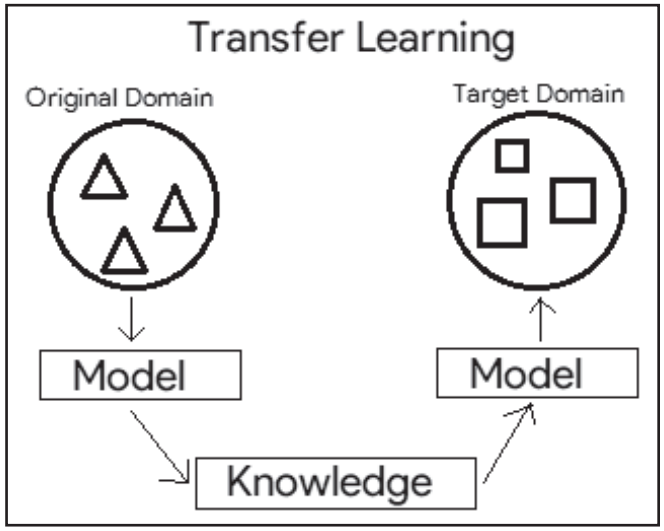
Manuscript received January 4, 2018; revised January 30, 2018; accepted February 6, 2018. Date of publication March 6, 2018.
* R. Rajkumar is with Department of Computer Science, SRM University, Chennai, India - 603203 (email: rajkumar03r @gmail.com)
** A. Kaushal is with Department of Computer Science, SRM University, Chennai, India - 603203 (email: arnav.kaushal800@gmail.com)
*** A. Saha is with Department of Computer Science, SRM University, Chennai, India - 603203 (email: aishik.me@gmail.com)
DOI: 10.17010/ijcs/2018/v3/i2/123212

from prior knowledge and human effort in feature design is a major advantage.

B. Transfer Learning on CNNs

A convolutional neural network built to solve a particular problem can be used in transfer learning to solve related problems. This is due to the architecture of the CNN [6]. It is a multilayer neural network having weights in between. According to the architecture, the lower level layers and weights store the most primitive knowledge. For example, in an image classification task the primitive knowledge refers to the recognition of edges and lines. As the layers progress, the knowledge stored by them also gets more complex [4]. The layers near the end perform most of the classification task. The last layer most commonly a Softmax layer holds the labels to our classification task. These labels have associated accuracy values after completion of the classification from an image. In our application to recognize objects we first delete the last softmax layer and attach our new layer with our new labels required for classification. Then we reinitialize the weights of the last couple of layers to randomized values. We then retrain the CNN. The retraining step for our new task takes much less time than the original as the primitive features are already learned. On testing with Google's Inceptionv3 model on a dataset of flowers, our transfer learning method took minutes to train. The original Inceptionv3 model took many days to train on computation heavy infrastructure.

Fig. 1. Overview of the transfer learning technique



III. HARDWARE FOR ACCELERATED MACHINE LEARNING

Most machine learning libraries and frameworks support the use of graphics processing units (GPUs) for performing the training process. For large organizations performing machine learning research and deploying applications there was a bottleneck due to the limited capability of GPUs in computing. To remove this drawback and help scale machine learning applications, the tensor processing units commonly referred to as TPUs were built. Google pioneered the development of TPUs and has successfully deployed them for its own use and is also making them available for the public through its cloud computing platform.

A. GPUs

A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. This has been extensively used for training deep neural networks [7] due to the enhanced speed of training compared to general CPUs. The CUDA is a parallel computing platform and application programming interface model created by Nvidia. The machine learning frameworks for training on the GPU uses it. Libraries like TensorFlow have support for CUDA.

B. TPUs

Tensor processing unit (TPU) [8] was announced by Google in 2016. It is used in their datacenters for computing results for their speech and AI platforms. Compared to a graphics processing unit, it is designed for a high volume of low precision computation, as little as 8-bit precision with higher IOPS (Input Output Operations per second) per watt, and lacks hardware for rasterization or texture mapping. Currently, two generations of TPUs have been released. Google has planned to provide access to their TPUs through their cloud platform.

IV. MODEL ARCHITECTURES AND SOFTWARE

Transfer learning can be applied to deep learning models including convolutional neural networks. We tested this procedure with two models, namely, the Inceptionv3 model and MobileNet model. Both models

were pre-trained on the ImageNet dataset to recognize objects. We used these models to train our flower classifier. We also used the TensorFlow Python library for our code.

A. Inceptionv3

This model was trained from scratch to its best performance on a desktop with 8 NVIDIA Tesla K40s in about 2 weeks [9]. It was trained for the ImageNet [5] Large Visual Recognition Challenge using data from 2012. This is a standard task in computer vision, where models try to classify entire images into 1000 classes. To apply transfer learning on this model we first collected our dataset. We used a dataset of five different flowers. First we removed the last layer of the Inceptionv3 model and appended our own layer for the classification of labels. Then we retrained the model, but before this step we passed all pictures from the dataset through the model and cached the resultant weights. These cached weights are called bottlenecks and are used in the training step. This is one reason that the training was accelerated because we did not find the primitive features again. The model generated after transfer learning is rather large for mobile devices to run because the weights are in large bit float values. This model can be quantized i.e. the weights can be rounded, to be able to run on mobile devices. However, without quantization this model can be deployed in the cloud for solving problems streamed from other devices.

B. MobileNet

MobileNet is a class of efficient models for mobile and embedded vision applications [10]. It is based on a streamlined architecture that uses depth-wise separable convolutions to build lightweight deep neural networks. These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. This model is also trained on the ImageNet Large Visual Recognition Challenge. As the architecture of both the Inceptionv3 and MobileNet models are similar and the last layer is the same, we were able to use the same training script on both the models. The model created by retraining MobileNet is small and can be deployed on mobile devices for on device offline object recognition. This is useful in many applications as will be mentioned in a later section.

C. TensorFlow

TensorFlow is an open source library developed by the Google Brain team [11]. It is used for dataflow programming across a range of tasks. It is used for both research and production at Google. It supports many languages like Python and Go. Developing deep learning models becomes an easier task because many great abstractions like Keras are found in this library. A tensor can be thought of as a large vector that passes through a graph that represents the nodes of the deep neural network.

V. APPLICATIONS AND BENEFITS

Transfer learning is considered the next driver for commercial machine learning success. It is highly beneficial in saving time and resources in computing. It has many applications.

A. Training Custom Object Classification Models

There are highly accurate models available that perform object classification like the Inceptionv3 and MobileNet models. Transfer learning allows us to train these models for our own custom dataset. For example, a dataset of flowers can be used as input for transfer learning to generate a flower classifier. This may be beneficial for students and professionals of botany and floriculture.

B. Scarce Dataset Problem

There are situations in which the dataset available has less data. This may be because of various factors such as high cost of data acquisition or difficulty in obtaining data. Some datasets pertaining to medical experiments are not available due to non-availability of permissions. This problem is solved when using transfer learning to solve a related problem which is already solved. As the learned data is same for the primitive layers of the deep neural network, lesser data is required during training. Time is also saved and this leads to an accelerated rate of experiment cycle. Thus, insights will be higher and may lead to better results. Medical applications of this method are highly in demand. Large amounts of labeled data are usually proprietary or expensive to obtain, as in the case of many speech or MT datasets, as they provide an edge over competition.

C. Diabetic Retinopathy

Diabetic Retinopathy is a complication of diabetes that affects the eyes. It affects up to 80% of people who have had diabetes for 20 years or more. At least 90% of new cases can be reduced if there is proper treatment and

monitoring of the eyes. Thus, it is essential to diagnose this situation as fast as possible. As a large dataset is available for this condition, we may use our Inceptionv3 model to find whether the retinal image has diabetic retinopathy or not. The created model may be quantized and deployed as an app on mobile devices. This will assist doctors and ophthalmologists to diagnose with ease.

D. Detecting Cancer in MRI Scans

The previous method for Diabetic Retinopathy can be extended to detect cancer in MRI scans. Even if a scarce dataset is available, a working model may be generated that can tell if any scan has cancerous behavior. This will help medical professionals diagnose patients and also provide decisions when the doctor cannot make a judgement.

E. Cloud Platform For Training Custom Models

Our transfer learning script may be used as the backend for a cloud platform that is used to train custom object classifiers. This platform may be extended to other domains like speech and text. Google launched Cloud AutoML, a platform where users can upload their datasets, models will automatically be created, and an API will allow them to access the classification model. Users can train high quality custom machine learning models with minimum effort and machine learning expertise.

VI. CONCLUSION

Thus, we see the power of transfer learning to accelerate the machine learning research cycle by shortening the training period. This technique also enables us to create models with good accuracy from scarce datasets. Coupling the transfer learning technique with advanced specialized hardware will enable researchers to derive more insights in less time. This is a path to democratize machine learning for non-professional developers who can use pre-built APIs to apply deploy machine learning on their applications.

ACKNOWLEDGMENT

The authors would like to thank their institution for patronizing their research endeavors and helping publish this work. Aishik Saha thanks his parents for their compassion and unwavering belief.

REFERENCES

[1] R. Rabiser and R. Torkar, "Guest Editors' Introduction : Special Section on Software Eng. and Advanced Appl.," *Inform. and Software Technol.* , 2015, vol. 67, pp. 236. DOI: https://doi.org/10.1016/j.infsof.2015.06.005

[2] L. Fei-Fei, and O. Russakovsky, "Anal. of large-scale visual recognition," Bay Area Vision Meeting, October, 2013.

[3] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. of Big Data*, vol. 3, no. 9, 2016. https://doi.org/10.1186/s40537-016-0043-6

[4] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Trans. on Knowledge and Data Eng.* , vol. 22, no. 10, pp. 1345-1359, 2010. doi: 10.1109/TKDE.2009.191

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. of the ACM, vol. 60, no. 6, pp. 84-90, 2017. DOI: https://doi.org/10.1145/3065386

[6] Q. Yang, "When Deep Learning Meets Transfer Learning," In Proc. of the 2017 ACM on Conf. on Inform. and Knowledge Manage. (CIKM '17)," ACM, New York, NY, USA, pp. 5, 2017. DOI: https://doi.org/10.1145/3132847.3137175

[7] X. Chen, J. Chen, D. Z. Chen, and X. S. Hu, "Optimizing memory efficiency for convolution kernels on Kepler GPUs," In *Proc. of the 54th Annu. Design Automation Conf. 2017 (DAC '17)*. ACM, New York, NY, USA, Article 68, 6 pages, 2017. DOI: https://doi.org/10.1145/3061639.3062297

[8] N. P. Jouppi et. al., "In-datacenter performance anal. of a tensor process. unit," *Proc. of the 44th Annu. Int. Symp. on Comput. Architecture*, 2017. DOI: 10.1145/3079856.3080246

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," Comput. Vision and Pattern Recognition (CVPR), 2016. DOI: 10.1109/CVPR.2016.308

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[11] Abadi et. al., "TensorFlow: A system for large-scale mach. learning," In *Proc. of the 12th USENIX Conf. on Operating Syst. Design and Implementation (OSDI'16)*, *USENIX Assoc. , Berkeley,* CA, USA, pp. 265-283, 2016.

About the Authors

R. Rajkumar is with Department of Computer Science, SRM University, Chennai, India.

Arnav Kaushal is with Department of Computer Science, SRM University, Chennai, India.

Aishik Saha is with Department of Computer Science, SRM University, Chennai, India.

INDIAN JOURNAL OF COMPUTER SCIENCE

Statement about ownership and other particulars about the newspaper "Indian Journal of Computer Science" to be published in the 2nd issue every year after the last day of February.

FORM 1V
(see Rule 18)

1.	Place of Publication	:	NEW DELHI
2.	Periodicity of Publication	:	BI- MONTHLY
3.	4,5 Printer, Publisher and Editor's Name	:	S. GILANI
4.	Nationality	:	INDIAN
5.	Address	:	Y-21,HAUZ KHAS, NEW DELHI - 16
6.	Newspaper and Address of individual	:	ASSOCIATED MANAGEMENT
	Who owns the newspaper and partner of	:	CONSULTANTS PRIVATE LIMITED
	Shareholder holding more than one percent.	:	Y-21, HAUZ KHAS, NEW DELHI-16

I, S.Gilani, hereby declare that the particulars given above are true to the best of my knowledge and belief.

DATED : 1st March, 2018

Sd/-
S. Gilani
Signature of Publisher