Transfer Learning and its application in Computer Vision: A Review

Chandeep Sharma, Sayonee Parikh
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
chandeep.sharma23@gmail.com
sayoneeparikh@gmail.com

***Abstract*—When it comes to data mining and machine learning algorithms a common assumption is made that the same feature space holds the training data as well as future data and also that it has similar distribution. In real world applications we might have a classification task and its training data in different domains of interest or the training data might have a different distribution of data or might be in a different feature space. Hence our initial assumption does not always hold true. In such scenarios the concept of transferring of knowledge can be of great benefit. Recently, transfer learning has become prominent as a new field which helps in addressing this problem. Knowledge transfer when executed in a proper manner is certain to improve the efficiency and will be cost effective as it will save us from expensive data labeling. This review purely focusses on transfer learning and transfer learning in the context of deep learning.**

Keywords—Transfer Learning, Deep Learning, Source and target Domain, Fine tuning, Convolution, Feature extraction.

I. Introduction

Computer Vision programs have been successful in achieving exceptional things with the use of Artificial Intelligence which prove to be a great benefit to society. Miraculous success has been endured by machine learning algorithms and data mining in several domains of knowledge engineering for example regression, clustering, classification etc. Majority of machine learning and AI models work on the acceptance that the testing and training data are obtained from the same feature space and or distribution. If any changes are made in the distribution space, there will be a need to build a brand-new statistical model from the base up by training it with newly obtained data. This process is time consuming and most certainly expensive in most real-world applications. It is also difficult, and in some cases almost impossible to collect new and relevant training data required for rebuilding the model. During such times if the need to re-collect the training data is reduced then that would be such a relief. This can be achieved with the help of *transfer learning*. It can also be simplified as *knowledge transfer.*

Transfer learning is basically transferring and leveraging the knowledge gained from one task and utilizing it to solve other related tasks. This in-turn helps in overcoming the isolated learning paradigm of the traditionally designed conventional Deep Learning and Machine Learning algorithms. Transfer learning can be more or less seen as a "design methodology" rather than a machine learning technique[11]. This is one of the most promising areas which could also lead to development of Artificial General Intelligence (AGI) , says the CEO of DeepMind *Demis Hassabis*. There are a large number of examples where the concept of transfer learning can be really advantageous. One such example would be that of WiFi-based indoor localization where the data can be easily outdated [12]. Here the aim is to detect a person's present location based on previously collected data of WiFi. In a large-scale environment, calibration of WiFi data is really expensive to build localization models because labeling of huge collection of data of the signals is required at each location. A model trained for a particular device or time period may reduce the performance for location estimation on different devices or time periods. Here to reduce the effort to recalibrate the model transfer learning could be beneficial to adapt the localization model trained in source domain for target domain.

In this paper we focus on the definition, Transfer Learning strategies, Transfer Learning for Deep Learning and some pretrained models for Computer Vision such as VGG-16, InceptionV3 and ResNet and finally we conclude the article.

II. Transfer Learning

A. *Traditional ML vs. Transfer Learning*

In traditional learning the tasks are generally isolated, and the learning is based on a single task without retaining any knowledge. The models are trained on that specific dataset. As there isn't any information retention, no knowledge can be transferred to the other models. On the other hand, with transfer learning the knowledge related to weights, features, labels etc. can be leveraged from models trained previously and newer models can be trained from the information obtained and the problems of not having enough data for training can also be tackled with this. The figure below explains traditional machine learning vs transfer learning in a nutshell.

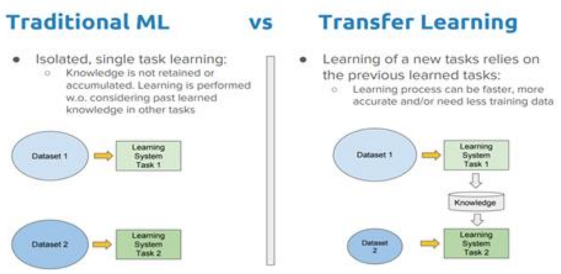

Fig 1[1]

B. *Definition*

In the paper, a Survey on Transfer Learning[13], to describe the framework to understand transfer learning, Pan and Yang use marginal probabilities, domain and tasks as a representation. It is defined as follows:

If we have a source domain A_s and a corresponding source task B_s, similarly a target domain A_T and task B_T, the function *of transfer learning is to improve the predictive function f of target domain by utilizing the knowledge in A_s and B_s. Here $A_s \neq A_T$ or $B_s \neq B_T$.*

Here the feature spaces and marginal probability distributions of the source domain and target domain are different. Also, the label spaces and conditional probabilities between the source task and target task are different.

C. Transfer Learning Strategies

Based on different tasks, domains we are dealing with and availability of data, there are different techniques and strategies[1] of transfer learning which can be applied.

1. Inductive Transfer learning: In this type of learning the target task differs from source task. Here the source domains and target domains could be the same or different. Source domain's inductive biases are utilized by the algorithms to improve the target task. This learning can be further divided into self-taught learning and multi-task learning depending on whether the source domain contains labeled data or unlabeled data.
2. Transductive Transfer learning: In this type of setting the source task and the target tasks are the same and the source domain and target domains are different. Here there will be plenty of labeled data available in the source domain, but no such data will be found in the target domain. A further classification can be done based on whether marginal probabilities are different from the feature spaces.
3. Unsupervised Transfer learning: This type of setting is particularly similar to inductive transfer learning. Here the focus is on unsupervised tasks in the target domain. The source domain and target domain are similar, and the source and target tasks are different. This setting lacks labeled data in either of the domains.

D. Transferable Components

The types of transfers that can be made across the categories[1] discussed above are as-

1. Transferring Knowledge of Instances: This approach of instance transfer to the setting of inductive transfer learning is quite desirable. Here the data from the source domain cannot be directly reused. However, some parts of the domain's data along with certain target domain's data can still be reused for the sake of improving results. Boosting algorithms such as TrAdaBoost by Dai et al.[14] are quite resourceful in utilizing instances from training of source domains causing improvements in the target tasks.
2. Feature-representation transfer: This approach aims at finding decent feature representations so that the error related to divergence of domain and/or classification and regression models can be minimized. Unsupervised or supervised methods can be used for feature representation transfer based on availability of labeled data.
3. Parameter transfer: In this type of transfer an assumption related to the model is made such as the model for the tasks related have some parameters in common or prior distribution of hyperparameters is shared. Here an additional weightage is applied to the target domain's loss for improvement in overall performance.
4. Relational-knowledge transfer: This type of transfer handles non-independent identically distributed data i.e., non-IID data which is not seen in the other three approaches discussed. The example of social network can be appropriate here which uses relational-knowledge transfer wherein each data point has a relationship with others.

D. Transfer Learning for Deep Learning

The idea of applying Transfer learning to Deep Learning could be understood from the analogy that Deep Learning models are representative of inductive learning[1]. To quote, in case of classification, the task of training models using labeled data to generalize well on unseen data involves the algorithm to learn about the distribution of the training data itself. The algorithm works with the underlying assumptions of the training data called inductive bias[2] where it recognizes the relationships or the trends existing between given input and output data.

Further, the idea of transfer learning here takes two pointers-

- Consider a trained model on a different domain with a different source task.
- Adapt it to the target domain with the target task.

Hence, Inductive transfer learning technique utilizes the source task available to assist with the target task, as from Fig D.1. The approach can be either by narrowing down search space, altering search process using source task information etc.

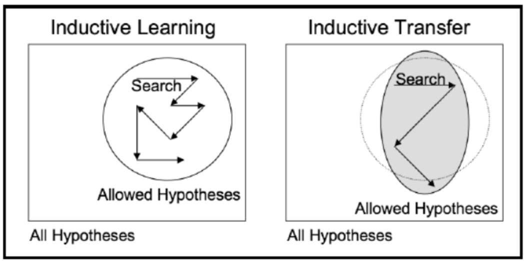

Fig 2[1]

E. Deep Transfer Learning Strategies

Popular strategies include-

1. Pre-trained models as feature extractors: The idea is to utilize one or more layers of an existing deep neural model without its final layer, on a different yet related task as general feature extractor. This way, the new model revamped would have to be trained in a shallow way, improving computation time and resources. Pre-trained model's weighted layers won't be updated with the unseen data, during back propagation.

   It assumes that the tasks are different but belong to the same domain.

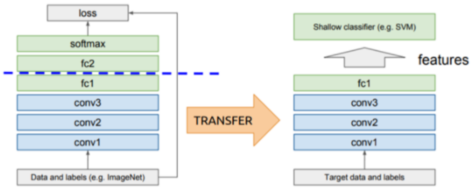

Fig 3[1]

2. Fine Tuning the pre-trained model: Unlike the previous, the layers from the pre-trained model are re-trained but in a shallow and selective way. Here, the states of the model are analyzed, where some of the weights are fixed[1] (freezed) while others are fine-tuned.

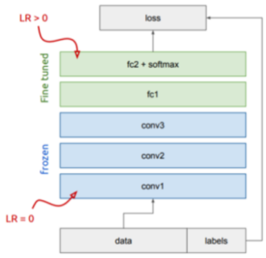

Fig 4[1]

F. *Types of Deep Transfer Learning*

Transfer learning in Deep Learning or Transfer Deep Learning, is a general concept where information/knowledge from existing source task is used to assist the target task, in the same or different domain. Hence its variations[1] include, but are not limited to-

1. Domain Adaptation: A well-known Transfer Learning approach is Domain adaptation[3], based on the scenario that source and target task domains hold different but related distributions. One such example is the spam filtering problem where a model from one user is applied to a new user, who receives a different set of emails.
2. Domain Confusion: To mention, layers in a deep neural model capture different set of features, some of which can be recognized as invariant. Such features can be transferred across domains, by tweaking the domain of target to arouse similarity to that of source, so as to apply the learned model, say by pre-processing[1].
3. Multitask Learning: Multitask Learning[4] aims at improving generalization by learning multiple tasks at the same time. It can be viewed as copying human learning capabilities, where knowledge is transferred from one task to another wherever these tasks seem related, as in Fig 5[1]
4. One-shot Learning: Aligning with real-world scenarios, where getting labeled data is not always possible, one-shot learning is a type of transfer learning where output is inferred using one or few training examples.
5. Zero-shot Learning: Extreme variant of Transfer learning[1] where unlabeled data is used to learn a task, analogous to unsupervised learning. It makes use of clever adjustments during training of the model, to understand patterns of unseen data. One such application is in Natural language processing, where translators are built using data with no labels of target language.

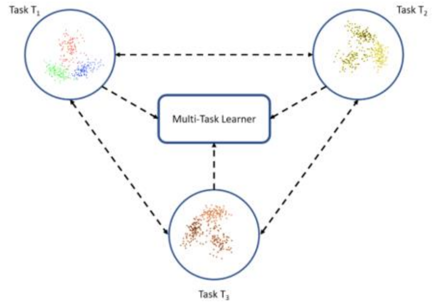

Fig 5[1]

III. PRE-TRAINED MODELS FOR COMPUTER VISION

Pre-trained models[1] are usually shared in the form of the millions of parameters/weights the model achieved while being trained to a stable state. There are several of them available for applications in Computer Vision as-

A. *VGG-16*

Visual Geometry group, abbreviated as VGG, derived from the group of researchers who developed it and 16 implies that its architecture has 16 weight layers. It is a deep convolution neural network that achieved 92.7% accuracy on ImageNet database with 14 mn images belonging to 1000 classes[7].

Architecture[6]*:* A fixed size input of 224x224 RGB image is fed to the 16 layer stack of conv. layers, where filters of 3x3 are used. A 1x1 conv. filter has also been used to introduce non-linearity of the input channels. The stride and padding is fixed to one pixel per filter, ensuring the preservation of spatial resolution. 5-2x2 max-pooling windows with stride 2 are added for spatial pooling. 3 fully connected layers in the stack are as: 4096 channels in the first two, 1000 channels for 1000 image classes in the third layer perform classification and finally the architecture has soft-max layer at last.

VGG16 is an improvement over AlexNet, where large kernel-sized filters are replaced by multiple 3x3 filters as a stack[5]. The architecture as a whole is as-

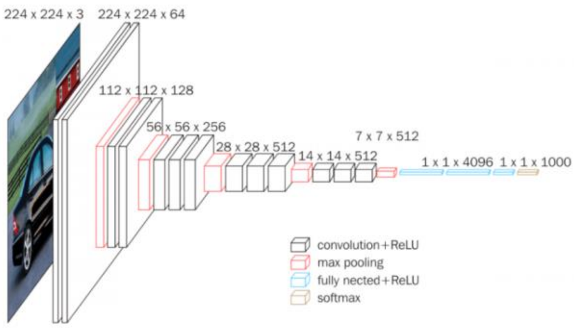

Fig 6. VGG-16 Architecture[5]

The model is explainable and achieves sufficient accuracy when testing on image data for classic classification tasks, but due its depth and large number of weight parameters, the model is large in size and has high inference time.

One of the variations to the model was introduced as VGG-19, where the depth of the network was pushed from 16-19 weight layers[8].

B. *ResNet*

This model introduces a residual learning framework to improve the training of deep neural networks, hence the word ResNet[9]. To quote the efficiency in the deep network, the degradation problem is targeted. The accuracy of a deep network increases with depth, it saturates and then degrades rapidly. ResNet uses residual mapping where instead of allowing every layer to fit a desired underneath mapping, the layers are allowed to fit a residual mapping. The building block is as[9]-

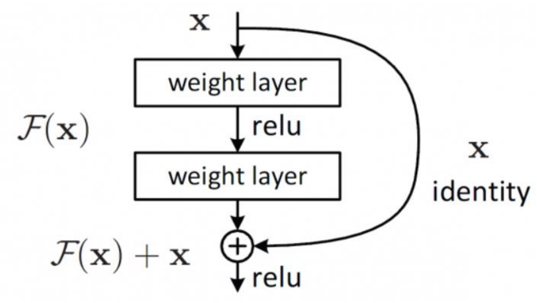

Fig 7. Building block of ResNet[9]

Architecture: As compared to VGG, the residual neural network is less complex with fewer filters and less amount of training. The shortcut connection is added that turns the network into its residual version[9], performing identity mapping, without padding hence with no additional parameters. Mathematically, the projection shortcut is F(x{W}+x).

ResNet is either two layers deep such as ResNet18 and 34, or three layers deep such as ResNet50, 101 and 152.

C. *Inception v3*

In order to increase efficiency and computational power of Inception networks (v1 etc), and easier model adaptation[9], techniques of factorized convolutions, regularization, dimension reduction and parallelized computations were implemented in v3.

Architecture: The model architecture can be explained in steps as[9]-

1. Factorized Convolutions- Keeps a check on network efficiency and computational efficiency by reducing the number of parameters.
2. Smaller Convolutions/filters- This leads to faster training as the number of parameters are reduced.
3. Asymmetricity- Simply replacing 3x3 with 1x3 and 3x1 convolution as-

Fig 8[9]

4. Auxiliary Classifier- Auxiliary classifier acts as a regularizer, CNN is inserted between layers during training.
5. Grid Size Reduction- Pooling operations are then added.

Consolidated architecture is as-

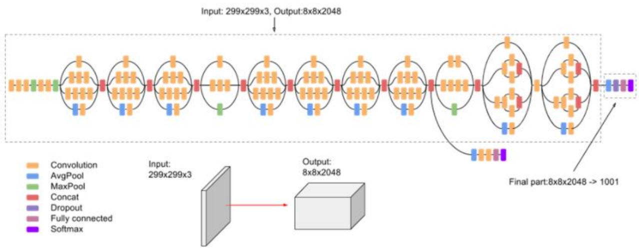

Fig 9[9]

The list for pre-trained models specifically for Computer Vision applications is long enough, and with each change in the network, comes better efficiency.

IV. CONCLUSION

Overall, through this paper we aimed to learn and review concepts about Transfer Learning, which is a machine learning technique, where a model massively trained on one task can be used to make predictions on other tasks, in a related domain. The transfer of knowledge takes place either through Feature extraction or Fine tuning of the pre-trained model.

Transfer Learning with predictive modeling problems can be widely used for Image data, as seen from the description on pre-trained models such as VGG, ResNet, Inception etc. Generally, the researchers who developed these models using efficient GPU resources, make them available online for use as pre-trained models. The approach is really effective as these models exist that are already trained on a large corpus of image data, with a relatively large number of classes.

However, given the advantages, it is not always the case that Transfer Learning would always give better accuracy on a particular task, their evaluation can be done only when they have been developed. While, it is useful in scenarios where there is not much data or even on a related task that has plenty of data, but the model can be re-used feasibly, avoiding building one from scratch.

At last, three important questions should be considered while applying the concepts of Transfer learning~ What, When and How to transfer. Next, the choices of source data or source model is an open problem and may require domain expertise.

References

[1] D. Sarkar, "A Comprehensive Hands-on Guide to Transfer Learning with Real-world Applications in Deep Learning", *Towards Data Science,* 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

[2] "Inductive Bias", *wikiwand.com*. [Online]. Available: https://www.wikiwand.com/en/Inductive_bias

[3] "Domain Adaptation", *wikipedia.com*. [Online]. Available: https://en.wikipedia.org/wiki/Domain_adaptation

[4] A. Sharma, "Understanding Multitask Learning with Transfer Learning". [Online]. Available: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15894480.pdf

[5] T.J. Perumanoor, "Introduction to VGG-16", *medium.com,* Sep 23, 2021. [Online]. Available: https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615

[6] F. Chollet, "Deep Learning with Python", *livebook.manning.com*. [Online]. Available: https://livebook.manning.com/book/deep-learning-with-python/about-this-book/

[7] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition". [Online]. Available: https://arxiv.org/abs/1409.1556

[8] K. Simonyan, A. Zisserman,"VGG-19", *kaggle.com.* [Online]. Available: https://www.kaggle.com/keras/vgg19/home

[9] V. Kurama, "A Review of Popular Deep Learning Architectures: ResNet, Inception V3, and SqueezeNet", *blog.paperspace.com*, 2020. [Online]. Available: https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/

[10] What is transfer learning? Exploring the popular deep learning approach. (n.d.) *Built In*. [Online]. Available: https://builtin.com/data-science/transfer-learning

[11] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for wifi-based indoor localization," *in Proceedings of the Workshop on Transfer Learning for Complex Task of the 23rd AAAI Conference on Artificial Intelligence,* Chicago, Illinois, USA, July 2008.

[12] Pan, S. J., & Yang, Q. (2009). "A Survey on Transfer Learning", *IEEE*, *22*(10), 1345–1359. [Online]. Available: https://doi.org/10.1109/TKDE.2009.191

[13] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in Proceedings of the 24th *International Conference on Machine Learning, Corvalis,* Oregon, USA, June 2007, pp. 193–200.