Improving Transferability of Deep Neural Networks

Parijat Dube, Bishwaranjan Bhattacharjee, Elisabeth Petit-Bois and Matthew Hill

Abstract Learning from small amounts of labeled data is a challenge in the area of deep learning. This is currently addressed by Transfer Learning, where one learns the small dataset as a transfer task from a larger source dataset. Transfer Learning can deliver higher accuracy if the hyperparameters and source dataset are chosen well. One of the important parameters is the learning rate for the layers of the neural network. We show through experiments on the ImageNet22k and Oxford Flowers datasets that improvements in accuracy in range of 127% can be obtained by proper choice of learning rates. We also show that the images/label parameter for a dataset can potentially be used to determine optimal learning rates for the layers to get the best overall accuracy. We additionally validate this method on a sample of real-world image classification tasks from a public visual recognition API.

Keywords Deep learning · Transfer learning · Fine-tuning · Deep neural network · Experimental

E. Petit-Bois—Work done when the author was a student intern at IBM Research AI.

P. Dube (✉) · B. Bhattacharjee · E. Petit-Bois · M. Hill
IBM Research AI, Yorktown Heights, NY 10598, USA
e-mail: pdube@us.ibm.com

B. Bhattacharjee
e-mail: bhatta@us.ibm.com

E. Petit-Bois
e-mail: epetitbo@students.kennesaw.edu

M. Hill
e-mail: mh@us.ibm.com

B. Bhattacharjee
Kennesaw State University, Kennesaw, GA 30144, USA

R. Singh et al. (eds.), *Domain Adaptation for Visual Understanding*,
https://doi.org/10.1007/978-3-030-30671-7_4

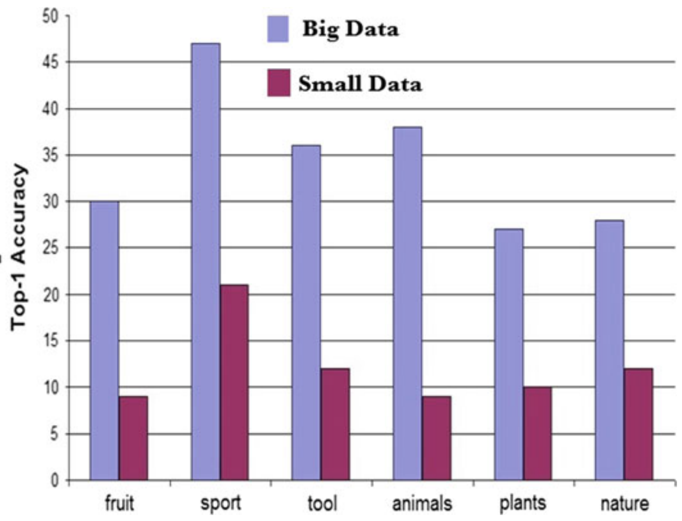

Fig. 1 Impact of data size on learning accuracy

1 Introduction

Deep Learning has become all pervasive in many application domains like Vision, Speech, and Natural Language Processing [13]. This can be partly attributed to the availability of fast processing units like GPUs as well as better neural network designs. The availability of large, open source, general-purpose labeled data has also helped the penetration of Deep Learning into these domains.

The accuracy obtained on a learning task depends on the quality and quantity of training data. As Fig. 1 shows, with larger amounts of data, for the same learning task, one can obtain much better accuracy. In this figure, the accuracy obtained on various categories of ImageNet22K [5] are shown with the big data being 10x bigger in size than the small data. While large, open source, general purpose, labeled data is available, customers often have specific needs for training. For example, a doctor may be interested in using Deep Learning for Melanoma Detection [4]. The amount of labeled data available in these specific areas is rather limited. In situations like these, the training accuracy can be negatively impacted if trained with only this limited data. To alleviate this problem, one can fall back on Transfer Learning [14, 17].

In Transfer Learning, one takes a model, trained on a potentially large dataset (called the source dataset) and then learns a new, smaller dataset (called the target dataset) as a transfer task (T) on it. This can be achieved by fine-tuning the weights of neurons in the pretrained model using the target dataset. Fine-tuning is a technique to leverage the information contained in a source dataset by tweaking the weights of its pretrained network while training the model for a target dataset. It has been shown that models trained on the source dataset learn basic concepts which will be useful in learning the target dataset [18].

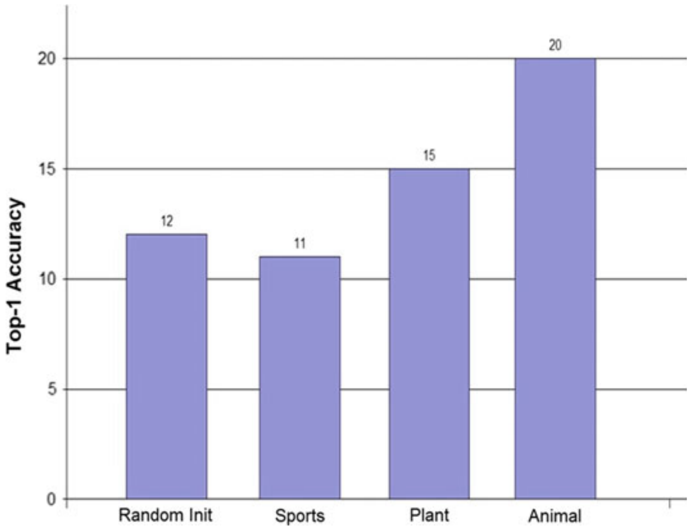

Fig. 2 Impact of base model on transfer learning accuracy

In the area of vision, the neural networks tend to be quite deep in terms of layers [10]. It has been shown that the layers learn different concepts. The initial layers learn very basic concepts like color, edges, shapes, and textures while later layers learn complex concepts [12]. The last layer tends to learn to differentiate between the labels supported by the source dataset.

The key challenges to Transfer Learning are how, what, and when to transfer [17]. One needs to address key questions like the selection of the source dataset, the neural network to use, the various hyperparameter settings as well as the type of training method to apply on the selected neural network and dataset. Figure 2 shows the accuracy obtained while training on the Tool category of ImageNet22K on models created from different source categories of ImageNet22K like Sports, Animals, Plant as well as random initialization. As the figure indicates, accuracy varied from -8% to $+67\%$ improvement over the random initialization (no Transfer Learning) case.

When performing Transfer Learning using deep learning, a popular method of training is using Stochastic Gradient Descent (SGD) [3]. In SGD, the key hyper-parameters to control the descent are the block size, the step size, and the learning rate. In the case of Transfer Learning, the learning rate can be set for every layer of the neural network. This controls how much the weights in each layer change as training progresses on the target dataset. A lower learning rate for a layer allows the layer to retain what it has learned from the source data longer. Conversely, a higher learning rate forces the layer to relearn those weights quicker for the target dataset. For Transfer Learning, the concepts learned in the early layers tend to have high value since the source dataset is typically large, and the early layers represent lower level features that are transferable to the target task. If the rates are large, then the weights could change significantly and the neural network could overlearn on the target task, especially if the target task has a limited amount of training data. The accuracy that is obtained on the target task depends on the proper selection of all these parameters.

In this chapter, we study the impact of individualized layer learning rates on the accuracy of training. We use a large dataset called ImageNet22K [5] and a small dataset called the Oxford Flowers [15] for our experiments. These experiments are done on a deep residual network [10]. We show that the number of images per label plays an important role in the choice of the learning rate for a layer. We also share preliminary results on real-world image classification tasks which indicate graduated learning rates across a network, such that early layers change slowly and allow for better accuracy on the target dataset.

The chapter is organized as follows: In Sect. 2, we describe related work. In Sects. 3 and 4 we describe our experimental setup and present our results, respectively. We conclude in Sect. 5.

2 Related Work

Several approaches are proposed to deal with the problem of learning with small amounts of data. These include one-shot learning [8], zero-shot learning [16], multitask learning [1, 7], and generic transfer learning [2, 9, 18].

Multitask learning simultaneously trains the network for multiple related tasks by finding a shared feature space [1]. An example is Neural Machine Translation (NMT) where the same network is used for translation to different languages [7]. In [9] a joint fine-tuning approach is proposed to tackle the problem of training with insufficient labeled data. The basic idea is to select a subset of training data from source dataset (with similar low-level features as target dataset) and use it to augment the training dataset for target task. Here, the convolutional layers of the resulting network are fine-tuned for both the source and target tasks. Our work is targeted for scenarios where source dataset is not accessible and fine-tuning is only possible using a target dataset.

It was established in [18] that fine-tuning all the layers of the neural network gives the best accuracy. However, there is no study on the sensitivity of accuracy to the degree of fine-tuning. In [2] it is experimentally shown for one dataset that the accuracy of a (fine-tuned) model monotonically increases with increasing learning rate and then decreases, indicating existence of an optimal learning rate before overlearning happens. We studied variation in accuracy of model with learning rate used in fine-tuning for several datasets and observed non-monotone patterns.

Another popular form of Transfer Learning is by using deep feature embeddings from a neural network to drive binary Support Vector Machines (SVMs) [2, 6]. In this approach, there are as many SVMs as categories in the target dataset and each SVM learns to classify a particular label. The feature embeddings can be taken from any layer of the neural network but, in general, is taken from the penultimate layer. This is equivalent of fine-tuning with the learning rate multipliers of all the inner layers up to the penultimate layer being kept to 0 and the last layer being changed.

3 Experimental Setup

ImageNet22k contains 21841 categories spread across hierarchical categories. We extracted some of the major hierarchies like sport, garment, fungus, weapon, plant, animal, furniture, food, person, nature, music, fruit, fabric, tool, and building to form multiple sources and target domains image sets for our evaluation. Figure 3 shows the hierarchies of ImageNet22k dataset that was used and their relative sizes in terms of number of images. Figure 4 shows representative images from some of these important domains. Some of the domains like animal, plant, person, and food contain substantially more images (and labels) than categories such as weapon, tool, or sport. This skew is reflective of real-world situations and provides a natural testbed for our method when comparing training sets of different sizes.

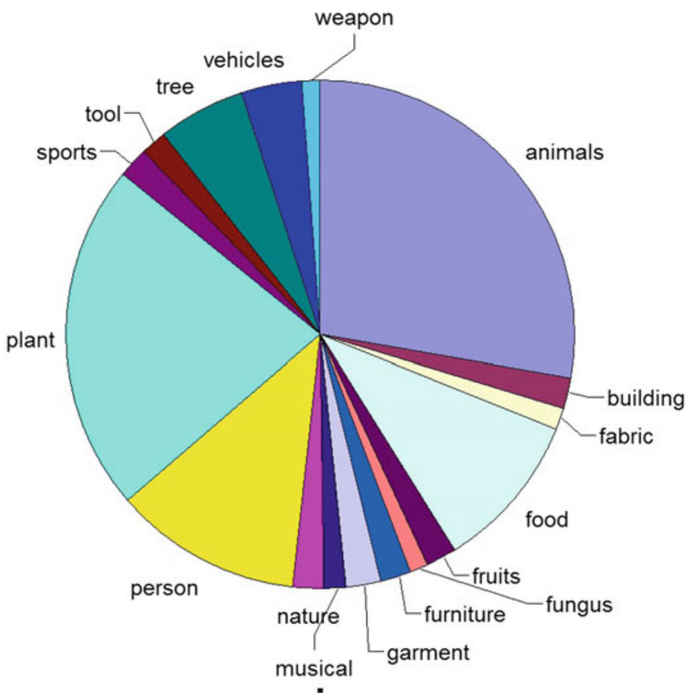

Fig. 3 Imagenet22k hierarchies used

Fig. 4 Representative images from various Imagenet22k hierarchies used in experiments

Each of these domains was then split into four equal partitions. One was used to train the source model, two were used to validate the source and target models, and the last was used for the Transfer Learning task. One-tenth of the fourth partition was used to create a Transfer Learning target. For example, the person hierarchy has more than one million images. This was split into four equal partitions of more than 250 K each. The source model was trained with data of that size, whereas the target model was fine-tuned with one-tenth of that data size taken from one of the partitions. The smaller target datasets are reflective of real Transfer Learning tasks.

We augmented the target datasets by also using the Oxford Flower dataset [15] as a separate domain. The dataset contains 102 commonly occurring flower types with 8189 images. Out of this, a target dataset of only 10 training images per class was used. The rest of the data was used for validation.

The training of the source and target models was done using Caffe [11] and a ResNet-27 model [10]. The main components of this neural network are shown in Fig. 5. The source models were trained using SGD [3] for 900,000 iterations with a step size of 300,000 iterations and an initial learning rate of 0.01. The target models were trained with an identical network architecture, but with a training method with one-tenth of both iterations and step size. A fixed random seed was used throughout all training.

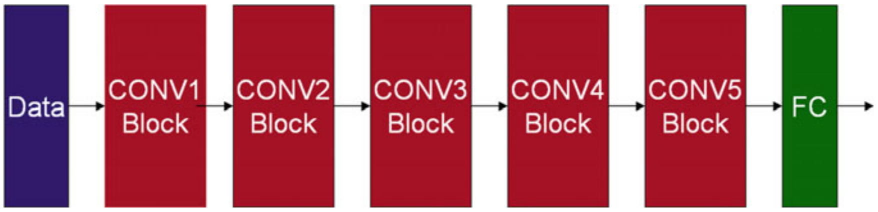

Fig. 5 Major Blocks of the ResNet model used in the experiments

4 Results and Discussion

Fine-tuning the weights involves initializing the weights to the values from the source model and then adjusting them to reduce the classification loss with the target dataset. Typically in fine-tuning a source model to a target domain, the practice is to keep the weights of all the inner layers unchanged and only fine-tune the weights of the last fully connected layer. The parameter which controls the degree of fine-tuning is the learning rate. Let $IL - n/LL - m$ be a transfer learning fine-tuning experiment where the inner layers learning rate (IL) is at n and outer layer learning rate (LL) is at m, with $n < m$. We are assuming a uniform learning rate for all the inner layers for most of the experiments. For those where the inner learning rate was varied, it is specifically mentioned in the chapter.

4.1 Fine-tuning Last Layer

We first did some experiments to quantify the gains possible by varying the learning rate of the last layer in fine-tuning while keeping all the inner layers weights unchanged. Table 1 compares the difference in accuracy of trained model for two different values of learning rate of the last layer, 0.01 and 0.1, corresponding to experiments $IL - 0/LL - 0.01$ and $IL - 0/LL - 0.1$. Observe that the accuracy is sensitive to the choice of LL and significant gains in accuracy (up to 127%) are achievable for certain domains by just choosing the best value of LL.

4.2 Fine-tuning Inner Layers

An earlier work [2, 18] has observed that fine-tuning inner layers along with the last layer can give better accuracy compared to only fine-tuning the last layer. However their observation was based on limited datasets. We are interested in studying how the accuracy changes with IL for a fixed LL with the following objectives:

Table 1 Transfer learning accuracy with varying LL

Target	Source	LL-0.01 (%)	LL-0.1 (%)	% Gain
Fabric	Garment	**13.09**	11.33	15.47
Tool	Weapon	14.54	**14.78**	1.63
Oxford	Plants	**91.06**	73.17	24.44
Food	Fruit	**5.71**	5.07	12.52
Fungus	Plant	**13.12**	5.80	127.79
Person	Food	**4.49**	2.81	59.75
Fruit	Garment	9.30	**10.50**	12.92
Music	Plant	**15.37**	9.47	62.22

(i) Identify patterns which can be used to provide guidelines for choosing LL and IL for a given source/target dataset.
(ii) Find correlation between dataset features like images per label, similarity between source and target datasets, and the choice of IL/LL.
(iii) Quantify possible gains in accuracy for different datasets by exploring the space of LL and IL values and hence establish the need to develop algorithms for identifying the right set of fine-tuning parameters for a given source/target dataset.

To this end, we conducted experiments varying IL for a fixed LL. We divided the experiments into two sets based on perceived semantic closeness of source and target domains. Set A (B) consists of experiments where the source and target datasets are semantically close (far). Thus we have,

$$\begin{aligned}A &= \{fabric_t/garment_s, tool_t/weapon_s, oxford_t/plants_s,\\ &\quad food_t/fruit_s, fungus_t/plant_s\}, \text{ and}\\ B &= \{person_t/food_s, fruit_t/garment_s, music_t/plant_s\}\end{aligned}$$

Figures 6 and 7 show the accuracy obtained by increasing IL by powers of 10 between 0 and LL for $LL = 0.01$ and 0.1. So when $LL = 0.01(0.1)$, IL took values in $\{0, 0.0001, 0.001, 0.01\}(\{0, 0.0001, 0.001, 0.01, 0.1\})$.

Two patterns across different experiments are observed: (i) accuracy increases monotonically with IL and then decreases (ii) accuracy alternates between increase and decrease cycles. The variation in accuracy with IL can be significant for certain datasets. Let min_m and max_m be the minimum and maximum value of accuracy obtained when IL is varied at $LL = m$ and β_m be defined as:

$$\beta_m = \frac{max_m - min_m}{min_m} \times 100 \tag{1}$$

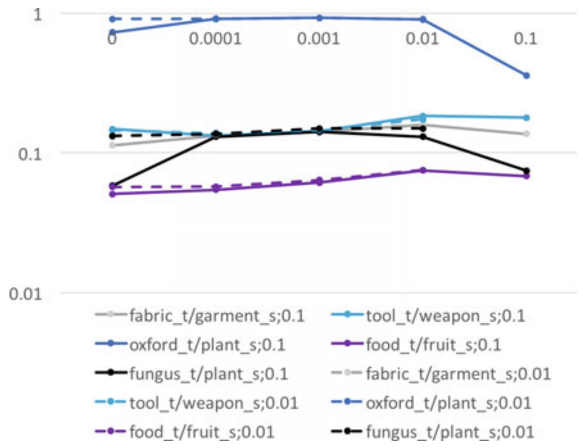

Fig. 6 Set A accuracy vs IL for fixed LL

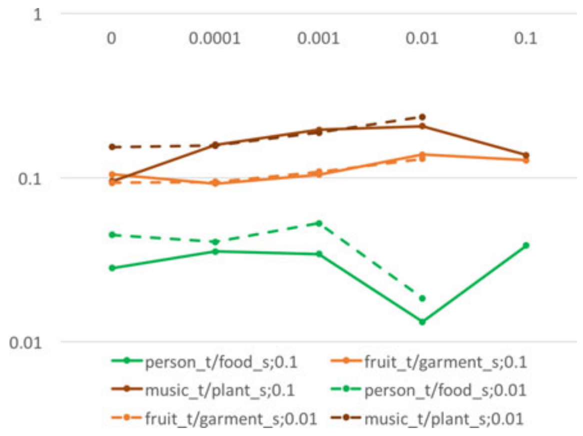

Fig. 7 Set B accuracy vs IL for fixed LL

Observe that β_m represents the percentage range of possible variation in accuracy with $LL = m$ and varying IL. Figure 8 compares β_m for different datasets. All the datasets exhibit $\beta_m > 0$, with median values of $\beta_{0.01}(\beta_{0.1})$ being 28.96% (83.52%). Observe that $\beta_{0.1} > \beta_{0.01}$ for all the datasets. Also, for same dataset, the range of variation in accuracy can be quite large or small depending on LL. For example, for $oxford_t/plant_s$, $fungus_t/plant_s$, and $music_t/plant_s$ the difference $\beta_{0.1} - \beta_{0.01}$ is greater than 100 points. Thus, fine-tuning both inner and outer layers gives the best accuracy. Further the value of IL that maximizes accuracy can be different for different datasets. The pattern of variation in accuracy with IL/LL is not always monotone.

Let α_m be the value of IL that achieves the best accuracy (max_m) at $LL = m$ for a dataset. Table 2 lists α_m for different datasets. The last column in the table shows the difference $max_{0.1} - max_{0.01}$. Observe that there is no clear winner, for some datasets keeping $LL = 0.1$ and then searching for IL gives the best accuracy while for others

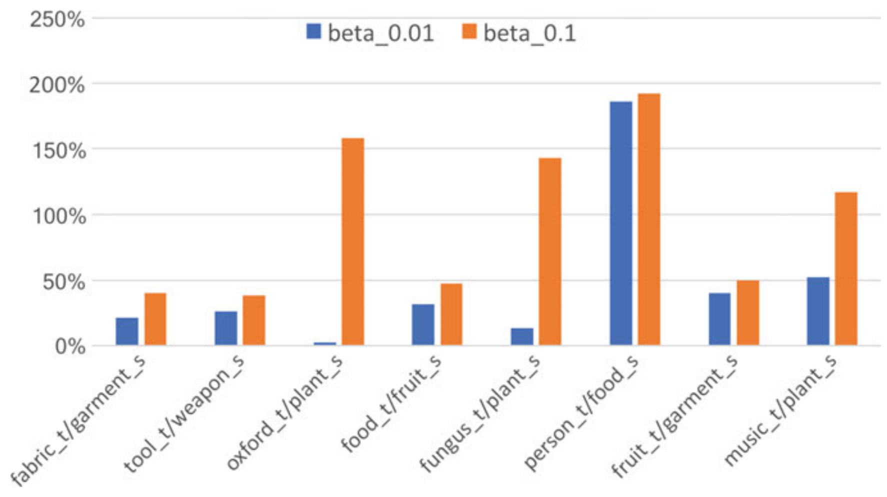

Fig. 8 Range of variation in accuracy with varying IL

Table 2 α_m for different datasets under study

Target	Source	$\alpha_{0.01}$	$\alpha_{0.1}$	$\max_{0.1} - \max_{0.01}$ (%)
Fabric	Garment	0.0001	0.0001	0.00
Tool	Weapon	0.0001	0.1	−1.41
Oxford	Plants	0.0001	0.0001	−0.88
Food	Fruit	0.01	0.01	0.98
Fungus	Plant	0.01	0.01	0.78
Person	Food	0.01	0.01	−0.71
Fruit	Garment	0.01	0.01	−0.12
Music	Plant	0.01	0.01	−2.86

$LL = 0.01$ performs better. This indicates the need for joint optimization over the space of LL and IL to get the best accuracy.

We are interested in identifying correlation between source/target dataset features and α_m. The first feature that we consider is images/label in the target dataset. Intuitively with more labeled data for the target domain, we can be more aggressive (i.e., use larger IL and LL) in fine-tuning. Figure 9 plots α_m versus images/label in target for $m = 0.01$ and 0.1. For both these cases we observe that α_m increases with images/label. However there is one anomaly, $\alpha_{0.1} = 0.1$ for $person_t/food_s$, though $person_t$ has smaller images/label. This seems to allude that other features of source/target datasets also dictate the choice of learning rates. We are currently investigating this direction with the hope to develop some functional mapping between the features of source/target datasets and α_m. This knowledge can be leveraged to develop intelligent algorithms to identify the best learning rate for inner layers and outer layers for a given source/target dataset.

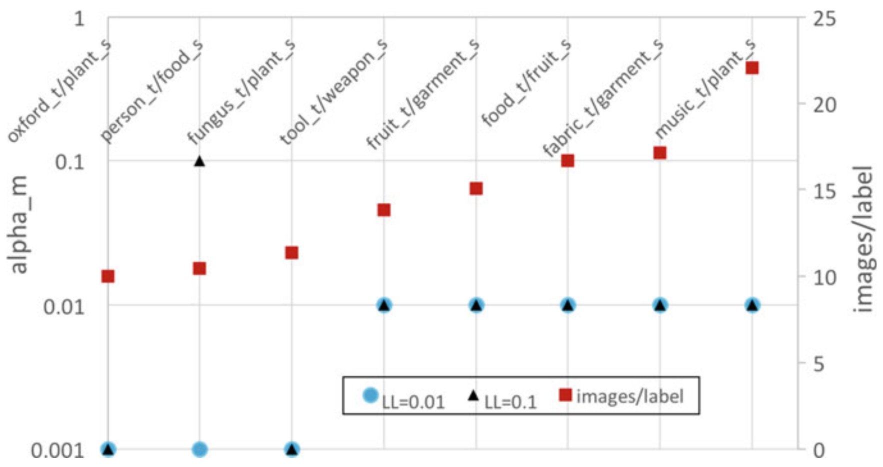

Fig. 9 Correlation between α_m and images/label

4.3 *Graduated Fine-tuning of Inner Layers*

We also investigated how the top-1 accuracy varies if the inner layer learning rate multipliers are not kept at a fixed value but varied. With the assumption that very basic concepts learned in the earlier layers are more important for transfer learning than later layers which map to complex concepts, we varied the learning rate multipliers in steps within the inner layers.

Oxford Flowers Dataset The ResNet-27 we are using for throughout these experiments has inner convolutional layers organized in five stages, conv1 through conv5 as shown in Fig. 5. We can denote the learning rate multiplier for each of these five stages as IL_1 through IL_5. We measured the accuracy of fine-tuning when we kept the inner learning rate multiplier ($IL_1..IL_5$) equal across stages, (at a fixed value of either 1, 2, or 5) and also compared to using a graduated set of values. In this case, each convolutional stage was assigned a multiplier (like 0, 1, 2, and 5), with conv1 and conv2 using the same (first, smallest) multiplier, and conv3, 4, and 5 using the successive, larger multipliers. (Meaning IL_1 was equal to IL_2.) In each case we set the learning rate multiplier LL of the last layer to 10. Figure 10 shows the top-1 accuracy for different IL configurations with Oxford flowers as the target dataset and plant as the source dataset with the base learning rate at 0.001. As the chart shows, the best accuracy was achieved when the learning rate multipliers were graduated.

Real-World Image Classification Tasks Next, we sought to validate these observations on training data “in the wild”. IBM operates a public cloud API called Watson Visual Recognition[1] which enables users to train their own image classifier by providing labeled example images, while images provided to the API are not used to

[1]https://www.ibm.com/watson/developercloud/visual-recognition/api/v3.

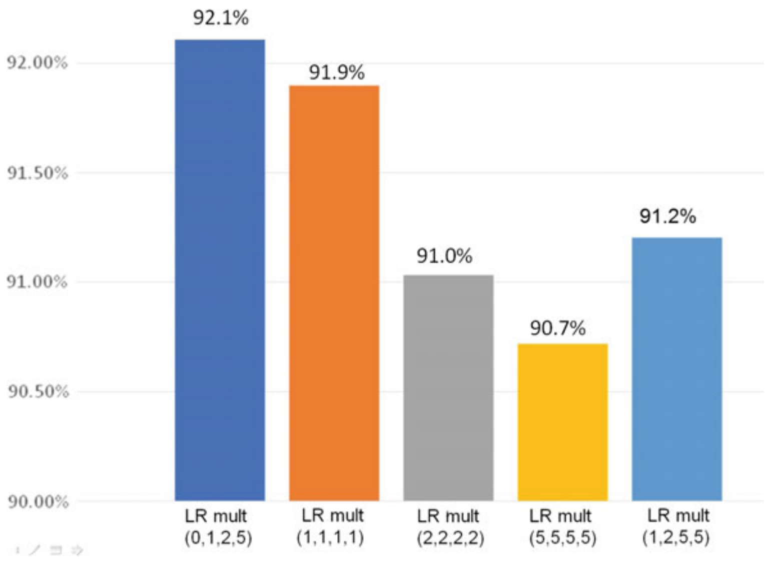

Fig. 10 Top-1 accuracy with varying inner LR mult and fixed outer LR mult at 20

train anything aside from that user's model. Users can opt-in to allow their image data to be used to help evaluate changes in the training engine. From the many training tasks that were opted-in, we took a random sample of 70 tasks. We did not manually inspect the images, but based on the names given to the labels, we presumed they represented a wide variety of image types, including industrial, consumer, scientific, and social domains as shown in Fig. 11. Based on the languages of the class labels, we had a wide geographic range as well. The average number of training images per task was about 250, with an average of 5 classes in each, so a mean of 50 image examples per class. We randomly split these into 80% for training and 20% for validation, leaving 40 training images per class on average.

For each of the 70 training tasks, we created a baseline model that was a ResNet-27 initialized with weights from an ImageNet1K model. We set the base learning rate to 0.001 and the LL to 10. The IL was set to 0. We fine-tuned the network for 20 epochs and computed top-1 accuracy on the held-out 20% of labeled data from each task. The average top-1 accuracy across the 70 tasks was 78.1%.

For the graduated IL condition, we initialized $IL_1..IL_5$ to be {0, 1, 2, 4, 8} and LL to be 16. We then defined a set of 11 scales, {0.25, 0.5, 1.0, 1.5, 2, 2.5, 3, 4, 5, 7, 10}. The scale is a secondary learning rate multiplier. For example, the final learning rate at scale 0.5 for conv3 (IL_3) and base learning rate 0.001 would be $0.5 * 2 * 0.001 = 0.001$. The intuition is to combine the scale factors explored in Figs. 6 and 7 with the graduated values of $IL_1..IL_5$ explored in Fig. 10.

This combination of scales and learning tasks resulted in $70 * 11 = 770$ additional fine-tuning jobs, which we ran for 20 epochs each. We evaluated the top-1 accuracy for each of these jobs. We found that if we picked the individual scale which maxi-

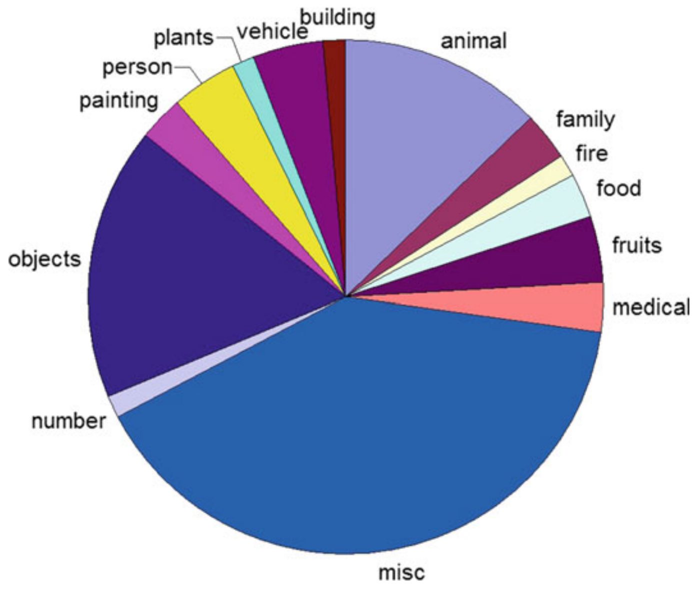

Fig. 11 Distribution of Image Classification Tasks from service API used

mized the accuracy for each job, the mean top-1 accuracy across all tasks improved from 78.1 to 88.0%, a significant gain. However, to find this maximum exhaustively requires running 11 fine-tuning jobs for each learning task. So we looked at which scale was most frequently the optimal one, and it was scale of 0.25. If we limit ourselves to one fine-tuning job per training task, and always chose this single scale, the mean top-1 accuracy across jobs had a more modest increase, from 78.1 to 79.7%.

This promising direction needs further investigation; if we could predict the optimal learning rate multiplier scale based on some known characteristic of the training task, such as number of images per class, or total number of training images, we could efficiently reach the higher accuracy point established by our exhaustive search.

5 Conclusion

Transfer Learning is a powerful method of learning from small datasets. However, the accuracy obtained from this method could vary substantially depending on the choice of the hyperparameters for training as well as the selection of the source dataset and model. We study the impact of the learning rate and multiplier which can be set for every layer of the neural network. We present experimental analysis based on the large ImageNet22K dataset, the small Oxford flower dataset and real-world image classification datsets and show that the images per label parameter could be used to determine what the learning rates. It also seems like continuously varying the learning rate for inner layers has more promise than keeping them all fixed and is a worthy direction to pursue.

References

1. Argyriou A, Evgeniou T, Pontil M (2006) Multi-task feature learning. In: Proceedings of the 19th international conference on neural information processing systems. pp 41–48
2. Bhattacharjee B, Hill M, Wu H, Chandakkar P, Smith J, Wegman M (2017) Distributed learning of deep feature embeddings for visual recognition tasks. IBM J Res Dev 61(4):1–9. https://doi.org/10.1147/JRD.2017.2706118
3. Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT
4. Codella N, Cai J, Abedini M, Garnavi R, Halpern A, Smith JR (2015) Deep learning, sparse coding, and svm for melanoma recognition in dermoscopy images. In: Proceedings of the 6th international workshop on machine learning in medical imaging, Vol 9352. Springer, New York, pp 118–126. https://doi.org/10.1007/978-3-319-24888-2_15
5. Deng J, Dong W, Socher R, Li LJ, Li K, FeiFei L (2009) Imagenet: a large-scale hierarchical image database. In: IEEE conference on CVPR
6. Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T (2014) Decaf: a deep convolutional activation feature for generic visual recognition. In: Proceedings of the 31st international conference on international conference on machine learning (ICML'14), Vol 32. pp I–647–I–655
7. Dong D, Wu H, He W, Yu D, Wang H (2015) Multi-task learning for multiple language translation. In: ACL
8. Fei-Fei L, Fergus R, Perona P (2006) One-shot learning of object categories. IEEE Trans Pattern Anal Mach Intell 28(4):594–611
9. Ge W, Yu Y (2017) Borrowing treasures from the wealthy: deep transfer learning through selective joint fine-tuning. In: Computer Vision and Pattern Recognition (CVPR)
10. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on CVPR
11. Jia Y, Shelhmer E, Donahue J, Kacayev S, long J, Girshick RB, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: ACM Multimedia
12. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. In: Neural Information Processing Systems
13. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
14. Mou L, Meng Z, Yan R, Li G, Xu Y, Zhang L, Jin Z (2016) How transferable are neural networks in NLP applications? In: EMNLP
15. Nilsback M, Zisserman A (2008) Automated flower classification over a large number of classes. In: ICVGIP
16. Palatucci M, Pomerleau D, Hinton G, Mitchell TM (2009) Zero-shot learning with semantic output codes. In: Proceedings of the 22nd international conference on neural information processing systems. pp 1410–1418
17. Pan SJ, Yang O (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng
18. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems 27 (NIPS 2014)