Towards Incremental Learning with Deep Convolutional Networks

Anuvabh Dutt

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble France
anuvabh.dutt@univ-grenoble-alpes.fr

RÉSUMÉ. *Les réseaux de neurones profonds sont des modèles d'apprentissage puissants. Cependant, ils requièrent du temps et des ressources importantes pour être entraînés. Nous proposons d'appliquer une approche d'apprentissage incrémentale pour entrainer ces réseaux en utilisant les informations présententes dans des modèles pré-entrainés. Nous souhaitons pour cela étudier les relations entre les architectures des réseaux, les catégories à apprendre, la quantité de données disponibles et la 'nature' de ces données. Nous présentons nos résultats sur l'effet des variations des architectures des réseaux en accord avec les données. Nous étudions l'entrainement de modèles dans des cas où le pouvoir discriminant d'un modèle doit être augmenté, y compris avec le même jeu de données sur lequel ils ont déjà été entraînés. Les résultats vont vers une architecture spécifique pour chaque tâche. Ce travail est réalisé dans le contexte de réseau convolutionnel et les performances sont mesurées en terme de précision pour la classification d'image*

ABSTRACT. *Deep neural networks are a powerful class of machine learning models. However they require a lot of time and computational resources to train. We propose to apply an incremental learning approach to train models by utilizing the information present in pre-trained models. We build towards this goal by studying the relationship between network architecture, categories in training data, the amount of training data, and the 'nature' of the data. We present our findings on the effect of varying network architectures with respect to the data. We investigate training models in scenarios where the discriminatory power of a model has to be increased, even for the same data set on which it has already been trained. The results are pointers towards having an optimal architecture for a specific task. The work is done in the context of convolutional neural networks and the performance is measured in terms of accuracy on an image classification task.*

MOTS-CLÉS : *l'Apprentissage en Profondeur, Indexation Multimédia Apprentissage Progressif Vision par Ordinateur*

KEYWORDS: *Deep Learning, Multimedia Indexing, Incremental Learning, Computer Vision*

1. Introduction

Human beings learn in different ways. Learning by observation and by building on existing knowledge is one of them. Humans try to draw parallels between what they need to learn, and what they already know. For humans, this happens quite naturally and intuitively. Artificial neural networks are an attempt at a simplified model of the neurons in a brain and in this work, we study how they can be made to learn, and build, from the *knowledge* that they already have. Here, knowledge is referred to the parameters of a neural network.

Recently, deep neural networks, trained in a supervised manner, have become the state-of-the-art machine learning models in several applications. Convolutional networks are a class of neural networks which are widely used in several computer vision tasks. These models enjoyed a lot of success in the nineties (LeCun *et al.*, 1998) and then fell out of fashion. This was primarily due to the lack of computing power needed to train deep models, and the unavailability of sufficient amounts of cleanly labeled data. With the recent advances in computing power, availability of a large labeled data set (Russakovsky *et al.*, 2015), convolutional networks have outperformed other techniques, in terms of performance on several benchmarks.

However, this performance comes at a cost. Training these networks require *computing resources* and *time*. There has been a lot of work where neural networks trained on one task, have been reused in another one (Sharif Razavian *et al.*, 2014 ; Yosinski *et al.*, 2014). This practice is generally referred to as *transfer learning*. It is quite common to use networks trained on large data sets, on other tasks, where the amount of labeled training data is relatively small. However, there is not much published work on understanding and increasing the 'capacity' of such models. Here, 'capacity' refers to how well a neural network model is able to distinguish, or classify, among several categories.

In this work, we study the behaviour of convolutional neural networks in an *incremental learning* scenario. We define incremental learning in two ways : 1) as the number of categories over which a model has to classify and 2) an increase in training data. We perform all of our work in the context of convolutional networks and evaluated their performance, in terms of accuracy, on an image classification task. The insights obtained will provide guidance on how a neural network architecture needs to evolve with the number of categories in a data set. Changes in network can include increasing the depth, increasing the number of neurons in a layer and splitting the connections among adjacent layers. We perform experiments to gain insights on how networks should be modified. This is important if we are to move towards generic network architectures.

Most of current research has focused on improving the performance of models on several benchmark data sets and reducing model size. The current procedure of reusing models, involve fine-tuning, which is sort of a trial and error process. It would be beneficial to know what exactly is needed, when the distribution of a data set, on which a model was trained, changes. Hence, we try to go towards that direction. A

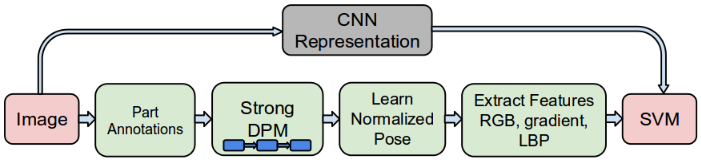

Figure 1 – CNN features used to obtain a generic image representation, which performs at par with the state of the art techniques, in several diverse computer vision tasks [1]. The bottom row pipeline consists of many manual labeling, and engineered feature transformations, such as SIFT, HOG.

better understanding of this would offer several benefits. It would lead to reduced training time, if a network can be 'expanded' from a pre-trained model. It would be enable a network to grow, as and when labeled data is available for new categories, again leading to a reduction in training time. It also leverages the current availability of pre-trained models, minimizing the need for training models from scratch.

2. Incremental Learning

One of the primary reasons that convnets are such a powerful class of machine learning models is that these models act both as feature extractors and as classifiers. This means that there is a reduced need for human engineered features, as is dominant in the case of most other machine learning models. Indeed, in most other models, *feature engineering is the trickiest part* and it plays a critical role in determining the performance of a model.

It was empirically shown in (Sharif Razavian *et al.*, 2014), that the features learnt by a neural network perform at par, or better, on several computer vision tasks, as compared to hand engineered features. This is depicted in figure 1. In their work, test images were fed into a convnet and the output of intermediate layers were extracted for each input. These outputs were treated as features and were used to train models such as support vector machines (SVM). The results obtained were at par, or better, than traditional highly tuned models. The remarkable thing is that the features were extracted from a network that was trained on a vastly different data set, as compared to the data sets on which the extracted features were used. This shows that a generic image representation was obtained when using features extracted from convnets. This is indeed a step towards an artificially intelligent system, as all of these features were learnt by the networks *without any manual intervention*. Deep neural nets require a lot of time and computational resources for their training. Hence, it is of interest to reduce both of these factors. (Yosinski *et al.*, 2014) performs a detailed experimental

[1] Figure taken from (Sharif Razavian *et al.*, 2014).

study on the transferability of features learnt by deep neural networks. Their results show that the shallower layers of a network are quite general and hence these weights can be used in diverse tasks. This provides a good starting point to train networks, when a pre-trained network is already available.

Since it quite evident that parts of a neural network model can be reused, we investigate the role of some of these parts in the context of incremental learning. One of the most important applications is in expanding a pre-trained model, when sufficient new labeled data is available for some concept classes. This is important to be able to avoid training from scratch a large model, when only a few concept classes have to be added as compared to the number of concept classes it is already trained upon. This scenario is surprisingly common. For example, in the case of the ImageNet data set (Russakovsky *et al.*, 2015), out of 22K concept classes, only 10K have sufficient number of training examples. As several models that are trained on these categories are available, a suitable incremental learning framework can help in expanding these models to newer concept classes, when enough data becomes available for them.

In the context of this work, we define *incremental learning* according to several aspects :

– The process of increasing the ability of a model to discriminate among an increasing number of classes. Specifically, let us that there is a neural network model that is trained on N concept classes. We investigate what is required of the model to enable it to discriminate among M concept classes, where $M > N$.

– The addition of more training samples, with and without changing the concept classes.

– Split existing concept classes to finer, more defined concept classes.

2.1. *Related Work*

Recent work has focused on improving the performance of convnets on several benchmark data sets (Simonyan et Zisserman, 2014 ; Szegedy *et al.*, 2015 ; He *et al.*, 2015), in utilizing pre-trained networks for diverse tasks (Girshick *et al.*, 2014), and on reducing model size (Han *et al.*, 2015). In another scenario big *teacher* models are used to train *student* models, with lesser parameters and gains in performance (Hinton *et al.*, 2015). In most of these scenarios, the number of target concept classes remain *constant or are lesser* in number compared to the pre-trained model. In this work, we study what characterizes the discriminatory power of a convnet when the number of target concept classes is *increased*.

Some recent work that has investigated the incremental learning problem are :

– Error driven incremental learning (Xiao *et al.*, 2014) proposes to increase network capacity by a clone and branch methodology to include new concept classes. However, they demonstrate their approach only on ‘animal’ classes.

– Mediated Experts for Deep Convolutional Networks (Agethen et Hsu, 2016) : This architecture proposes to have several networks trained on different concept classes. Some of the layers of these networks are shared, since it has been shown that lower layers learn generic features (Yosinski *et al.*, 2014). This is as effort to reduce the size of the overall model, which is an ensemble of networks. A confidence module is proposed, whose task is to reduce computation, in the case that one of the networks has very low "activations" for a certain input sample.

A mediator network is trained on all of the concept classes. This seems to defeat the purpose of an incremental learning methodology as it is implied that there is always access to all of the data. In their work, the results shown are for only two different concept classes.

In the above papers, there is an assumption that the data on which the available model (the pre-trained model) was trained is available. In the context of this work, there is no such assumption. Indeed, in most scenarios, it is unlikely to be able to have access to the data on which a different model was trained.

2.2. *Proposed Approach*

In an incremental scenario, where we want to increase the number of classes that a model is able to discriminate between, it is important to be able to observe which parts of the network are responsible for which task. We focus on two aspects, the network architecture and the training data. We outline next, the specifics explored, the motivation behind those investigations, and finally how we define the experiments to investigate each particular aspect.

Network Width

With respect to the network architecture, we explore the effect of the number of filters in each layer, on the network's ability to discriminate among classes. *The number of filters in a layer is referred to as the network width.* Networks with varying widths are trained in order to see the effect on model performance, for different number of concept classes. This also provides a look into how the network width relates to the number of target concept classes. The insight gained will provide pointers on how the network width should be changed, to enable a network to increase its discriminatory power.

Size of Training Set

After exploring the effect of changing the network width, we investigate the relation between network width and the amount of training data. This is of crucial importance when new (and sufficient) training data is available for concept classes that we would like to add to the network. Labeled data is expensive to obtain and hence it is crucial to be able to do the best possible.

Hierarchical Relationship Among Data

Another perspective of looking at incremental learning is through classification of coarse to fine grained classes. This is with respect to refining the target concept classes. For example, data sets like ImageNet have a hierarchical structure. We explore the relation between using networks trained on different levels of this hierarchy, with respect to weight initialization schemes, by exploiting these hierarchical relationships.

This situation arises, when a concept class is split into several sub-categories. This can happen, when training data is available for these sub-categories. This also happens when pre-trained models are used for different tasks. For example, the ImageNet data set has several "dog" categories, but in PASCAL VOC, there is just one "dog" category. When reusing models between such tasks, it is important to see how the model behaves.

3. Refining Concept Classes

In this section, we investigate the relationship between the network architecture and the number of concept classes. We refer to refining concept classes as an increase in the number of concept classes that a model has to discriminate over. This increase can be due to addition of new concept classes to the data set or due to splitting of existing concept classes into finer categories.

CIFAR Dataset Description

The CIFAR data set (Krizhevsky et Hinton, 2009) consists of 60000 images, each having dimension of 32×32 pixels. There are two distinct data sets, CIFAR-10 and CIFAR-100. Both of them have 50000 training images and 10000 testing images. CIFAR-10 has images belonging to 10 mutually exclusive classes. CIFAR-100 has images belonging to 100 different mutually exclusive classes and each of these 100 classes are grouped into 20 super classes. The 100 classes are denoted as *'fine classes'* and the 20 super classes as *'coarse classes'*. We will call models trained on these data sets as 'fine models' and 'coarse models' respectively.

Neural Network Architecture

As the goal is to investigate incremental learning, and not obtain the best possible performance, we chose an architecture from (Springenberg *et al.*, 2014), which is simple but has a good performance. It is described in table 1. The layer names are given in the first column of the table. A few points to note are :

– 'N' is the scale factor for the *network width*. It denotes the number of filters in a particular layer.

– 'C' denotes the number of concept classes among which the network has to classify.

– `pool3` is a global average pooling layer, with the number of output units equal to the number of concept classes in the data set.

Layer Name	Layer Type	Number of Outputs
data	Input 32 × 32 RGB image	
conv1	5 × 5 conv.	N ReLU
pool1	3 × 3 max-pooling stride 2	
conv2	5 × 5 conv.	2N ReLU
pool2	3 × 3 max-pooling stride 2	
conv3	3 × 3 conv.	2N ReLU
conv4	1 × 1 conv.	2N ReLU
conv5	1 × 1 conv.	C ReLU
pool3	global average pooling layer	C
fc	softmax with output units equal to C	C

Tableau 1 – Network architecture used for the experiments.

3.1. *Effect of the Network Width*

Filters are at the core of a convolutional network architecture. The intuition is that these filters *learn* to detect features. Each layer of the network computes a feature map by applying filters to the output of the previous layer. The number of filters in a layer in called the *network width* for that layer. Throughout the network, such feature maps are computed, and at the end the network makes a prediction. It is these feature maps which form the *feature extractor* part of a convolutional network. We investigate the effect of the number of filters in each layer on the performance of the model. The objective is to see how the performance is related to the number of concept classes in the data set.

The architecture from table 1, with different values of N, is used. N is in the range $\{32, 48, 64, 96, 128, 192, 256, 384\}$. The range of N values were chosen to include the ones most commonly used in practice, with some values bigger and smaller than those. After training, each model is evaluated on the corresponding CIFAR test set.

Figure 2 shows the accuracy of models, as compared to the number of filters and the number of concept classes in a data set. This shows that an increase in the number of filters helps in improving performance. Initially the gain is quite rapid but then the performance asymptotes. The surprising result is that the optimal number of filters is the *same*, irrespective of the number of concept classes among which the network has to classify. It seems that having more labels, does not translate into more information, which the network can extract. The performance of the model asymptotes at $N = 96$ for 10, 20 and 100 concept classes. Additionally, an increase in the number of filters does not lead to a drop in performance. It would seem that the filters learnt are robust to having additional filters, and are not negatively affected. There is no overfitting, which may be the effect of dropout. We conclude that just simply increasing the number of filters does not seem to fetch gains in performance. However it affects the computation time and hence it is important to choose the correct number of filters in

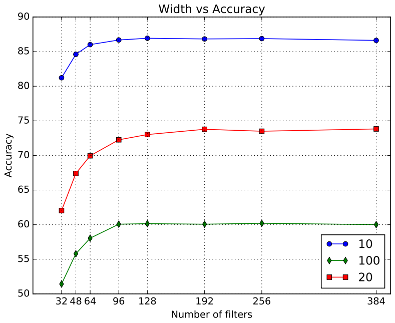

Figure 2 – Effect of the number of filters on the accuracy of models on CIFAR-10, CIFAR-100 fine and CIFAR-100 coarse data sets.

each layer.

3.2. *Effect of Training Set Size on Model Performance*

It is known that more the number of training samples, the better a machine learning model can generalize to new unknown examples. We investigate this, in an incremental learning scenario. The aim is to observe the relation between the *number of filters* in a convolutional network, the *number of concept classes* that the network has to classify, and the *number of training examples*.

Out of the 50000 training images, 10000, 20000 and 40000 training images are selected. This is done by uniformly sampling for the entire training set, with the requirement that all categories have the same number of training samples. This means there are 500, 1000, 2000 images per concept class for CIFAR-100 coarse, and 100, 200, 400 images per concept class for CIFAR-100 fine. Networks with N={32, 48, 64, 96, 128, 192, 256} are trained on these reduced data sets and classification accuracy is measured on the corresponding test set. Figure 3 show the performance of training on a reduced number of training examples for both the ‘coarse’ and ‘fine’ labels of CIFAR-100. Unsurprisingly, the more training examples we have, the better performance we obtain.

However, surprisingly, we find that if there are fewer training examples, increasing the width of the layers, *does help*. even if to a small extent. In figure 3, green and blue lines, corresponding to reduced number of training examples, show a greater increase in performance as compared to the red line. This is a pointer on how a network needs to be changed when additional training data is available. The implication is that increasing the number of parameters (filters), leads to gains in performance.

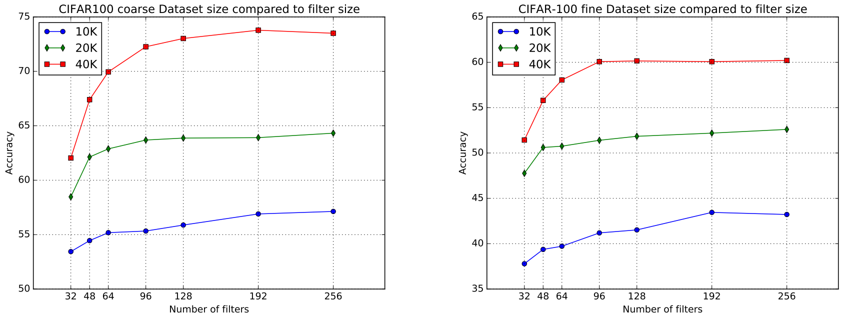

Figure 3 – CIFAR-100 : Effect of the size of data set, as compared to number of filters. 10, 20, 40 correspond to the the number of examples in the training set.

4. Conclusion and Perspectives

In conclusion :

– We investigated the relation between the network width and the number of categories that the network has to discriminate over. We find that the width helps to obtain a gain in performance only upto a certain extent. Further increase has no effect on the model performance. We also find that for an architecture, the 'optimal' network width is not related to the number of categories in the data.

– The relation between the width of a network layer, the size of the training set, and the number of concept classes that it contains, is investigated. We find that the width is important in obtaining a better performance when training data is scarce. This is an unexpected and counter intuitive result as we would expect that we could learn less parameters (weights) with less training data. However we see that we need more weights to get the best performance when less training data is available.

Future Work

– *Layer Activations to Guide the Learning.* For the same input image, the activations are quite different for networks trained on different data sets. The activations in different layers give clues about how each layer is contributing towards the performance of the network. For instance, if a layer has very low activations for a data set, it might mean that the "discriminatory" power of that particular layer needs to be increased. This is an area that can be investigated.

– *Sort training examples according to confidence of trained network.*. Curriculum learning (Bengio *et al.*, 2009) asserts that the order in which training samples are shown to the learning system influence the manner in which the system learns. Using the layer activation values, an ordering of training data might be determined for the expanding the capacity of the neural network. A reasonable definition of the confidence of a trained model, on new data, is needed to progress further. One idea is to use the variance of the softmax outputs.

Acknowledgement

This work is being carried out under the supervision of Georges Quénot and Denis Pellerin. This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

5. Bibliographie

Agethen S., Hsu W. H., « Mediated experts for deep convolutional networks », *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, p. 2687-2691, 2016.

Bengio Y., Louradour J., Collobert R., Weston J., « Curriculum learning », *Proceedings of the 26th annual international conference on machine learning*, ACM, p. 41-48, 2009.

Girshick R., Donahue J., Darrell T., Malik J., « Rich feature hierarchies for accurate object detection and semantic segmentation », *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 580-587, 2014.

Han S., Pool J., Tran J., Dally W., « Learning both weights and connections for efficient neural network », *Advances in Neural Information Processing Systems*, p. 1135-1143, 2015.

He K., Zhang X., Ren S., Sun J., « Deep residual learning for image recognition », *arXiv preprint arXiv :1512.03385*, 2015.

Hinton G., Vinyals O., Dean J., « Distilling the knowledge in a neural network », *arXiv preprint arXiv :1503.02531*, 2015.

Krizhevsky A., Hinton G., « Learning multiple layers of features from tiny images », 2009.

LeCun Y., Bottou L., Bengio Y., Haffner P., « Gradient-based learning applied to document recognition », *Proceedings of the IEEE*, vol. 86, n° 11, p. 2278-2324, 1998.

Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M. *et al.*, « Imagenet large scale visual recognition challenge », *International Journal of Computer Vision*, vol. 115, n° 3, p. 211-252, 2015.

Sharif Razavian A., Azizpour H., Sullivan J., Carlsson S., « CNN features off-the-shelf : an astounding baseline for recognition », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, p. 806-813, 2014.

Simonyan K., Zisserman A., « Very deep convolutional networks for large-scale image recognition », *arXiv preprint arXiv :1409.1556*, 2014.

Springenberg J. T., Dosovitskiy A., Brox T., Riedmiller M., « Striving for simplicity : The all convolutional net », *arXiv preprint arXiv :1412.6806*, 2014.

Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A., « Going deeper with convolutions », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 1-9, 2015.

Xiao T., Zhang J., Yang K., Peng Y., Zhang Z., « Error-driven incremental learning in deep convolutional neural network for large-scale image classification », *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, p. 177-186, 2014.

Yosinski J., Clune J., Bengio Y., Lipson H., « How transferable are features in deep neural networks ? », *Advances in neural information processing systems*, p. 3320-3328, 2014.