Abstract

Convolutional neural networks (CNN) have been shown to achieve state-of-the-art performance in a significant number of computer vision tasks. Although they require large labelled training datasets to learn the CNN models, they have striking attributes of transferring learned representations from large source sets to smaller target sets by normal fine-tuning approaches. Prior research has shown that these techniques boost the performance on smaller target sets. In this paper, we demonstrate that growing network depth capacity beyond pre-trained classification layer along with careful normalization and scaling scheme boosts fine-tuning by creating harmony between the pre-trained and new layers to adjust more to the target task. This indicates pre-trained classification layer holds high-level (global) image information that can be propagated through the newly introduced layers in fine-tuning. We evaluate our depth augmented networks following our designed incremental fine-tuning scheme on several benchmark datasets and show that they outperform contemporary transfer learning approaches. On average, for fine-grained datasets we achieve up to 6.7% (AlexNet), 5.4% (VGG16) and for coarse datasets 9.3% (AlexNet), 8.7% (VGG16) improvement than normal fine-tuning. In addition, our in-depth analysis manifests freezing highly generic layers encourage better learning of target tasks. Furthermore, we have found that the learning rate for newly introduced layers of depth augmented networks depend on target set and size of new layers.

1. Introduction

The origin of CNN dates back to early 1980s when Fukushima invented Neocognitron [9]. Later CNNs went out of spotlight due to lack of computation power (e.g. GPU) and sufficient labelled training data to avoid overfitting in supervised learning. The return of CNNs was paved by Krizhevsky et al. [18] in 2012 ILSVRC. They utilized both ImageNet dataset [6] and NVIDIA’s CUDA for reducing runtime [33]. Inspired by the success of AlexNet several CNNs are invented that are deeper and stronger in architecture [32, 13, 34]. CNNs still require huge amount of labelled training data to yield optimal performance. Luckily, training CNNs on a large and diverse enough dataset help them to transfer features across a wide range of tasks [28, 1, 43]. ImageNet acts as the source of humongous amount of labelled data. Feature-representation transfer and parameter transfer [25] are by far the top two transfer learning techniques used by deep learning community. The former transfers off-the-shelf CNN features from base to target task and the latter transfers the learned parameters. Both of these techniques assist training on the target task with limited labelled data and increase performance of the target model over random initialization. Transfer learning works well when the learned features are generic to both target and base tasks [43]. Transferring learned features can be utilized for target tasks by training classifier (e.g. SVMs) [5] as well. Sharif et al. [30] also trained SVMs with transferred features from ImageNet pre-trained CNNs for other computer vision tasks.

Hinton et al. [14] introduced fine-tuning for the first time by transferring knowledge from a generative to discriminative model. Later this approach was exploited with success by [12, 10, 44, 42, 24, 29, 7]. The basic sequence behind fine-tuning is to replace the last classification layer of an ImageNet pre-trained network with a randomly initialized new classification layer as per the target task. Then the modified new model goes through forward-backward propagations to tune gradient descent on the target set. Although almost every contemporary research prunes classification layer to follow basic transfer learning pipeline [36, 15, 21, 7], Azizpour et al. have trained external classifier (SVM) with the extracted outputs of pre-trained classification layer of AlexNet. They reported that this layer’s features are only for tasks that are subset of ImageNet semantic labels. This layer’s features are also reported to be efficient in classifying medical images by transfer learning [35]. Moreover, Azizpour et al. [1] reports increasing depth (new convolution layers) of network before the first fully-connected (FC)
layer performs better transferring of knowledge for both the distant and similar target tasks compared to fixed networks. On the other hand, Wang et al. [41] demonstrated growing new layers after second FC layer help the adaptation of pre-existing layers for target tasks. The following are the main contributions of this paper.

• Wang et al.’s [41] developmental learning motivated us to increase network depth for better fine-tuning. In contrary to [1], we employ pre-trained classification layer in the fine-tuning process and observe that this layer is efficient for improving classification performance of both similar and dissimilar target sets to ImageNet as shown in Figure 1 (row 2). So, we hypothesize that the presence of this layer (1000 × 4096 dimensional parameters) is vital to increasing depth of network because they carry high order image-level information. Therefore, we consider pre-trained classification layer as the last FC layer and append new layers beyond pre-trained classification layer. We find careful normalization is essential to create coordination between pre-trained and new activations. Moreover, we observe that augmenting multiple layer can further boosts up performance.

• To investigate optimal fine-tuning of our depth augmented network, we design an incremental layer-layer fine-tuning scheme. We systematically demonstrate that increasing the depth of network beyond pre-trained classification layer benefits transfer of knowledge to new tasks for both fine-grained and coarse datasets. Our depth augmented networks also outperform contemporary fine-tuning approaches having different network architectures e.g. fixed capacity network, depth or width augmented network.

• We present various interesting findings from our empirical study e.g. fine-tuning the whole transferred network except only highly generic layers is observed to be optimal, for new layers learn rate has to be greater than global and that is dependent to target set and size of new layers.

The rest of the paper is organised as follows. Section 2 discusses the related research in transfer learning. Details of our proposed depth augmented networks and incremental layer fine-tuning scheme for evaluating performance are presented in Section 3. Section 4 presents the study and analysis of our experimental results. Section 5 summarizes our contributions.

2. Related Work

Several works have experimented with developmental and lifelong learning [31, 4, 37, 26] which is in line with our approach to grow network capacity and learn. Our depth augmented network differs from models [38, 39] which increase network for memorizing new target sets. In contrary to contemporary research [41, 1], we grow network depth beyond classification layer (FC8 for AlexNet and FC16 for VGG16). Yosinsky et al. [43] show fine-tuning highly generic layers learn same generic features from target set that does not harm performance with similar source-target sets but harms dissimilar source-target sets. Their dissimilar split has natural versus man-made objects which does not portray correct visual dissimilarity. We plan to investigate their claim by our incremental layer wise fine-tuning scheme for both fine-grained and coarse datasets. Tajbakhsh et al. [35] have an incremental layer fine-tuning scheme which has a different layer update process than our proposed incremental layer fine-tuning scheme.

After implementing both feature and parameter transfer based learning with ImageNet pre-trained AlexNet Girshik et al. [10] made a curious claim to reduce the learning rate during fine-tuning to avoid “clobbering the initialization” of the pre-trained CNN. On the other hand, Nicholas et al. [2] reports global learn rate for transferred layers is optimal. For growing networks by augmenting new FC layers, they [41] have trained their newly augmented layers with 10 times more learn rate than other layers. We investigate their claim for increasing learn rate and report different outcomes. In the next sections, we will explain the architecture, methodology and target sets to address several gaps in the literature for transfer learning with depth augmented deep networks.
3. Depth Augmented Networks

This section presents the details of our proposed depth augmented network for optimal fine-tuning.

3.1. Architecture Overview

Let us assume that the pre-trained network representation module $X_P$ consists of $N$ layers including the classification layer. $L_N$ denotes $N^{th}$ layer with hidden activation $H^N \in \mathbb{R}^{Nk}$ layer $L_N$ has $Nk$ units. Let $W^N$ be the weights and $B^N$ be the biases of $N^{th}$ layer. That is, $H^N \leftarrow f(W^N H^{N-1} + B^N)$, where $f(\cdot)$ is a non-linear function, such as ReLU [22]. The new classifier module is denoted as $C$. We increase the depth capacity of the network by constructing a new FC layer $L_{N+1}$ having $S$ neurons on top of classification layer $L_N$. We evaluate our growing network in several different combinations where $S \in \{512, 1024, 2048, 4096 \}$. This depth augmented network is denoted by $X_D$ as shown in third row of Figure 1.Appending new layers create new activations, let us assume $H^{N+1} \leftarrow f(W^{N+1} H^{N} + B^{N+1})$ in layer $L_{N+1}$ is the new representation which is fed into $C$. In line with [41], appended layer is considered to be an adaptation layer which allows for suitable compositions of pre-existing parameters and avoid unwanted modifications to the pre-trained layers for their adaptation to the new task. Moreover, we increase network depth beyond layer $L_N$ by multiple new FC layers ($L_{N+1}$ and $L_{N+2}$) as shown in the fourth row of Figure 1. As multiple layers have more capacity to learn from scratch on the target task, we hypothesize that it will allow pre-trained parameters to adapt only necessary tuning without introducing noisy modifications. For growing multiple layers we experiment with the same set $S$ for each layer in various combinations.

3.2. Normalization Scheme

For establishing harmony between the learning of new and pre-trained parameters and reducing sensitivity to random initialization, we introduce batch normalization and adaptive scaling scheme in line with [16] to proposed networks. This is motivated by a recent work [19] that uses a combination of multi-scale pre-trained CNN features from several layers. Normalization and scaling scheme is placed in between fully-connected layers and non-linear functions. We put normalization scheme after pre-trained classification layer and newly appended layers for our evaluation. Normalization scheme first normalizes the activations $H$ of each channel by subtracting the mini-batch mean $\hat{H}$ and then dividing by the mini-batch standard deviation $\sigma$ as follows.

$$H' = \frac{H - \hat{H}}{\sigma}$$ (1)

Finally after shifting the input by a learnable offset $\beta$ the scheme scales input by a learnable scale factor $\gamma$.

3.3. Parameter Fine-tuning

All new parameters of layers $L_{N+1}$ and $L_{N+2}$ are randomly initialized during fine-tuning. Parameter fine-tuning is similar to training CNNs, during fine-tuning the following cost function is minimized with respect to the pre-trained weights $W$:

$$\mathcal{L} = -\frac{1}{|I|} \sum \ln(p(y^i|I^i)) + \frac{\lambda}{2T} \sum ||W||^2$$ (2)

where $|I|$ denotes the number of training images in the target set, $I^i$ represents the $i^{th}$ training image which has the corresponding label $y^i$. The probability is denoted by $p(y^i|I^i)$ for classifying $I^i$. Stochastic gradient descent (SGD) [3] is used for minimizing cost function, while fine-tuning the cost over the entire training set is approximated with the cost over mini-batches. $W_1^N$ denotes the weights of $N^{th}$ layer at $t^{th}$ iteration and $\mathcal{L}$ is the cost function over mini-batch of size $M$, then parameters updated for next iteration is as follows:

$$\Upsilon_t = \Upsilon_{[t M/||I||]}$$

$$\nabla_{t+1} = \mu \nabla_t - \Upsilon_t \delta \mathcal{L}$$

$$W_{t+1}^N = W_t^N + \nabla_{t+1}$$

where, $\alpha^N$ is the learning rate of $N^{th}$ layer, $\Upsilon$ denotes the scheduling rate for decreasing learn rate at every 10 epochs. The contribution of previous parameters update in the current iteration is represented by momentum $\mu$.

Let, $X_t \leftarrow F_T(Train\ set\ ,\ L_N \rightarrow C)$, where $X_t$ is the fine-tuned network and $F_T(\cdot)$ is a function which fine-tunes successive layers of $X_P$ and $C$ starting from layer $L_N$. To fine-tune parameters and evaluate our depth augmented networks, we have followed our designed incremental layer fine-tuning scheme stated in Algorithm 1. For each loop-iteration, we initialize the parameters with pre-trained or depth augmented CNN, fine-tune layers and record test accuracy. More specifically, only the last FC layer is tuned in the first iteration. The next iteration tunes the last and penultimate FC layers. This procedure to incrementally add the previous layer in the tuning is repeated until all the layers in the network is tuned. Note that in every iteration, we initialize with pre-trained CNN to make fair comparisons of the outcomes. After all layers are tuned, the inner loop terminates. We iterate every fine-tuning for 2000 iterations with a batch size of 100 and momentum 0.900. A global learn rate of 0.005 is used with a piece-wise scheduler which lowers down the learn rate to 10 times less than the previous one at every 10 epochs. To avoid overfitting while tuning, L2-regularization is used by setting the value of $\lambda = 0.004$. 

3
Algorithm 1: Incremental Layer Fine-tune.

Input:
- $I$: Labeled training images
- $J$: Labeled test images
- $X_D$: Depth augmented pre-trained network module
- $N$: Number of layers in $X_D$
- $L_N$: Denotes $N^{th}$ layer
- $C$: Classification module

Output:
- $X_t$: Fine-tuned network
- $TA$: Test accuracy on model $X_T$

Functions:
- $X_t \leftarrow F_T(I, L_N \rightarrow C)$ \{Fine-tune $L_N$ to successive layers only\}
- $TA \leftarrow X_t(J)$ \{Test accuracy on fine-tuned model $X_t$\}

1: Append $C$ with $L_{N-j}$
2: for $i = 0$ to $N - 1$ do
3: \quad $X_t \leftarrow X_D$
4: \quad $X_t \leftarrow F_T(I, L_{N-i} \rightarrow C)$
5: \quad $TA \leftarrow X_t(J)$
6: \quad $i \leftarrow i + 1$
7: end for

Parameter fine-tuning proceeds from $L_{N-1}/L_N$

Figure 2: Performance analysis of optimal fine-tuning from convolution layer 3 for AlexNet and convolution block 3 for VGG16.

Figure 3: Performance analysis of SVM approach as reported by [1].

4. Performance Study And Analysis

This section describes the datasets used in our experiment, the details of our experimental setup and our evaluation outcomes. We assembled a target dataset of two different fine-grained and two different coarse datasets. We have used ImageNet as the source dataset. Fine-grained datasets used in this work are 102 Flowers [23] having 102 categories of different flowers with 8189 images and CUB 200-2011 [40] having 200 types of birds with 11788 images in total. The Coarse or mixed semantic datasets used are Caltech-256 [11] with 30607 images having 256 categories and Pascal VOC-07 [8] having 20 different classes with 9963 images. A split of 75% is used for training and rest 25% for testing. The selected fine-grained sets have semantic labels far away from those used for optimizing the pre-trained CNN while the coarse sets have semantic labels that are subsets of source dataset. Selection of these target datasets is inspired from the interest to investigate the effect of parameter fine-tuning in proposed depth augmented networks for datasets that have similar and dissimilar semantic composition compared to source dataset. ImageNet trained $N^{th}$ layer’s output is only effective for datasets that are semantic subsets of ImageNet [1]. This finding intrigued us to investigate whether parameter fine-tuning with depth aug-

| Type      | Source          | Images   | Categories |
|-----------|-----------------|----------|------------|
| Fine grained | 102 Flowers | 08189     | 102        |
|           | CUB 200-2011   | 11788    | 200        |
| Coarse    | Caltech-256    | 30607    | 256        |
|           | PascalVOC 07   | 09963    | 020        |

Table 1: Selected datasets for target task.
Layer from which parameter fine-tuning proceeds
Top-1 accuracy

AlexNet Pre-trained Network

Figure 4: Accuracy graph of incremental layer fine-tuning of proposed depth augmented network and other approaches. Here, $L_N + C$ denotes Transfer learning with fine-tuning having $N$ layers. Depth augmented networks significantly outperform normal fine-tuning. Besides classification by using SVM with output of $N^{th}$ layer [1] lags far behind our proposed approach. Best performance is obtained when initial layers are frozen.

Figure 5: Accuracy graph of incremental layer fine-tuning of proposed depth augmented network and other approaches. Here, $L_N + C$ denotes Transfer learning with fine-tuning having $N$ layers. Depth augmented networks significantly outperform normal fine-tuning. 1, 2, 3, 4 and 5 in x-axis denote convolution blocks and present average results of layers within each block. For convolution blocks incremental fine-tuning evaluation is executed layer wise as stated in Algorithm 1. Note that different conv layers within block 3 yielded best results for different target sets.

Table 2: Improvement in accuracy (%) of proposed depth-augmented network with fully-connected classification layer from transfer learning with fully-connected classification layer and SVM [1]

| Dataset         | $L_{N-1}$ | $L_N$ | $L_{N-1}$ | $L_N$ |
|-----------------|-----------|-------|-----------|-------|
| CUB 200-2011    | 2.3       | 9.4   | -0.1      | 1.8   |
| 102 Flowers     | 4.8       | 3.8   |           |       |
| VOC-07          |           |       |           | 4.8   |

4.1. Comparison With Contemporary Approaches

Results shown in Figures 2 and 3 present that fine-tuning with layer $N$ is more efficient than utilizing external classifier (SVM). Table 2 shows performance improvement of fine-tuning with layer $N$ is higher compared to SVM approach. Therefore, we increased depth after $N^{th}$ layer and fine-tuned. For comparing our proposed depth augmented networks with normal fine-tuning, we evaluate them using our designed incremental fine-tuning scheme. Outcomes of experiments with new layers $L_{N+1}$ and $L_{N+2}$ consisting 2048 and 1024 neurons respectively are shown in Figure 4 and 5. We observe that networks of two different depth (AlexNet and VGG16) outperform normal fine-tuning for all selected datasets when new layers are appended. In addition, Figure 2 presents more clear view of significant improvements by our depth augmented networks. For fine-grained datasets, our proposed approach outperforms normal fine-tuning by 6.7% for AlexNet and 5.4% for VGG16 in average ash shown in Table 3. In the case of coarse datasets which have more semantic similarity with the source set, the performance increases by 9.3% for AlexNet and 8.7% for VGG16 than normal fine-tuning. This indicates that increasing depth beyond $N^{th}$ layer encourages better learning of new tasks. To further validate
Table 3: Improvement in accuracy (%) of depth-augmented networks at depth $N, N + 1, N + 2$ from the same at depth $N - 1, N, N + 1$, respectively, where parameter fine-tuning proceeds from $L_3$.

Table 4: Comparison of our depth augmented fine-tuned AlexNet module with contemporary fine-tuning approaches.

Table 5: Comparison of our depth augmented fine-tuned VGG16 module with contemporary fine-tuning approaches.

our claim, Table 4 and 5 summarizes classification performance comparison with different existing transfer learning techniques from literature. The best outcomes among various combinations of our depth augmented AlexNet and VGG16 evaluated by our incremental fine-tuning scheme are shown. For other approaches the performance gap created between our implementation and that reported by them is due to different target sets, train-test splits, network architectures and iterations. Note that we have used similar hyper-parameters, iterations and train-test splits for all approaches and Table 4 and 5 to maintain a fair comparison. Results show that our approach to grow network depth capacity outperforms other baselines. These empirical outcomes prove our claim of the presence of pre-trained classification layer in increasing model capacity is effective for adjusting network to target tasks. To better understand the robustness of our approach, diagnostic experiments having the number of augmented units for AlexNet and VGG16 are shown in Table 6. The diagnostic outcomes show that all of these variations of depth augmented networks significantly outperform normal fine-tuning. We evaluate our approach in all target sets, only two of them are depicted here for simplicity. The results on other target sets also consistent to this trend. The performance increases until the size of augmented layer is 2048 whereas it shows a marginal diminution when it reaches to 4096. Fine-tuning without initial convolution layers perform better than other cases.

4.2. Normalization Is Essential

To evaluate the claim by [41] that additional scaling is beneficial but inessential, we execute diagnostic experiments with FC8-1024 combination as shown in Table 7. The results for CUB indicate that without normalization followed by scaling scheme, the improvement of our depth augmented network is around 2% compared to normal fine-tuning, and more than 6% otherwise. Similar significant
| Network | Dataset          | Configuration | $L_N$ | $L_{N-1}$ | $L_{N-2}$ | $C_2$ | All  |
|---------|-----------------|---------------|-------|-----------|-----------|-------|------|
| AlexNet | CUB 200-2011    | FC8-1024      | 66.2  | 67.5      | 68.9      | 69.9  | 68.4 |
|         |                 | FC8-2048      | 66.3  | 68.7      | 69.0      | 70.4  | 69.2 |
|         |                 | FC8-4096      | 66.7  | 67.1      | 68.1      | 68.1  | 66.8 |
| CUB 200-2011 | FC8-1024      | 75.9          | 78.3  | 78.9      | 81.9      | 80.3  | 80.3 |
|         |                 | FC8-2048      | 75.2  | 77.9      | 78.5      | 82.1  | 81.1 |
|         |                 | FC8-4096      | 75.0  | 78.1      | 78.3      | 82.0  | 80.9 |
| Caltech-256 | FC8-1024      | 75.5          | 78.2  | 78.6      | 89.5      | 79.1  | 79.1 |
|         |                 | FC8-2048      | 75.7  | 77.9      | 78.5      | 82.1  | 81.1 |
|         |                 | FC8-4096      | 75.9  | 78.1      | 78.3      | 82.0  | 80.9 |
| VGG16   | CUB 200-2011    | FC16-1024     | 76.4  | 76.8      | 77.1      | 78.9  | 77.6 |
|         |                 | FC16-2048     | 76.8  | 77.1      | 77.5      | 78.6  | 78.8 |
|         |                 | FC16-4096     | 75.9  | 77.7      | 78.0      | 78.5  | 77.4 |
|         | Caltech-256    | FC16-1024     | 76.4  | 76.8      | 77.1      | 78.9  | 77.6 |
|         |                 | FC16-2048     | 76.1  | 76.7      | 77.9      | 79.1  | 78.5 |
|         |                 | FC16-4096     | 75.9  | 77.7      | 78.0      | 78.5  | 77.4 |
|         |                 | FC8-1024      | 85.5  | 88.1      | 88.7      | 91.1  | 88.5 |
|         |                 | FC8-2048      | 85.9  | 88.3      | 88.9      | 91.3  | 90.3 |
|         |                 | FC8-4096      | 85.3  | 88.1      | 89.5      | 91.9  | 90.9 |

Table 6: Layer wise fine-tuned depth augmented networks. Here, C and CB denotes convolution layer and block respectively. Here, FC8-2048 denotes new layer with 2048 neurons appended after FC8.

... performance boost noticed in other datasets after increasing network depth is fuelled by batch normalization. Therefore, we claim that normalization is essential for depth augmented networks and only naively appending layers do not always help network to adapt better.

4.3. Multiple Layer Augmentation

Appending multiple new layer after classification layer yielded even better results than our growing scheme with single new layer which clearly indicates that increasing network capacity incrementally helps network to generalize more on target sets. Note that this incremental depth augmentation requires careful normalization in between newly appended layers. It is proven once again that pre-trained 1000-way classification layer holds prominent high level features which are capable of propagating learned knowledge to multiple newly appended layers. Table 8 shows a summary of our multiple layer augmentation experiments for FC8-2048-1024 combination. That shows increasing network incrementally by augmenting depth is a stable parameterization for improving performance.

4.4. Deep Inside Fine-tuning

Our fine-tuning scheme for the proposed depth augmented network and the fixed capacity network depicted in Figures 4 and 5 initiate several interesting findings. Our empirical results prove Azizpour et al.’s [1] claim (i.e. FC8 features are only efficient for target sets which are subsets of ImageNet’s semantic labels) to be sub-optimal. We observe that FC8 features when fine-tuned with depth augmented layer outperform the normal fine-tuned FC6 and FC7 features. This signifies that fine-tuning with sufficient network capacity paves the way to learn better for layers that do not possess rich pre-trained features. Therefore, presence of pre-trained classification layer for increasing network depth is highly important for fine-tuning tasks for both subsets and not subsets of ImageNet semantic labels. Another intriguing finding is that fine-tuning the whole network except the highly generic layers (which hold gabor filters or colour blobs as stated by [43]) performs the best among all other combinations. This observation differs from their [43] claim that tuning FC1 and FC2 do not harm transfer learning. The intuition is that fine-tuning these generic layers might introduce noisy or unwanted modifications to the module. That is updating parameters would force the network to learn target sets generic features which are already learned from source set. Fine-tuning procedure has much less data and iterations than training from scratch which might not let huge number learned parameters of highly generic layers find another such equilibrium. Figure 4 portrays that for all selected target sets, starting fine-tuning from the third convolution layer yields the highest test accuracy. Apparently, Figure 5 also holds this similarity for convolution block 3. This gives an interesting insight that convolution blocks 1 and 2 of VGG16 might hold generic features or features that are highly in equilibrium because tuning them drops the performance.

4.5. Feature Visualization

We plot tSNE depending on the semantic category to understand the interrelation of learned features by implementing standard tSNE algorithm [20]. Figure 6 shows feature space of the normal fine-tuned FC7 (4096 dimensional) layer, FC7 (4096 dimensional), FC8 (1000-dimensional) and FC9 (2048 dimensional) layers of our depth augmented AlexNet. Apparently, Figure 6(a) shows normal fine-tuning improves a bit of semantic separation of the pre-trained network while FC7 features (Figure 6(b) tuned in our approach demonstrates significantly more clear semantic clusters because of increasing network capacity. Layer FC8 (Figure. 6(c)) and FC9 features (Figure. 6(d)) exhibit even better semantic cluster which help the proposed module to catch prominent features for classification from fine-tuning and...
| Network | Dataset         | Configuration          | Accuracy (%) |
|---------|-----------------|------------------------|--------------|
|         |                 | FC8-2048-1024          | 70.9         |
|         |                 | FC8-2048               | 70.4         |
| AlexNet | CUB 200-2011    | FC8-2048-1024          | 95.9         |
|         | 102 Flowers     | FC8-2048-1024          | 95.1         |
|         | Caltech-256     | FC8-2048-1024          | 82.9         |
|         |                 | FC8-2048               | 82.1         |
|         | VOC-07          | FC8-2048-1024          | 87.9         |
|         |                 | FC8-2048               | 87.1         |
| VGG16   | CUB 200-2011    | FC8-2048-1024          | 80.1         |
|         | 102 Flowers     | FC8-2048-1024          | 97.1         |
|         | Caltech-256     | FC8-2048-1024          | 92.5         |
|         |                 | FC8-2048               | 91.9         |
|         | VOC-07          | FC8-2048-1024          | 95.4         |
|         |                 | FC8-2048               | 94.9         |

Table 8: Comparison between growing network depth capacity by appending single and multiple layers. Only best combination is presented here.

4.6. Analyzing Learning Rate

Wang et al. [41] report that they increase the learn rate of newly appended layer by 10 times more than global rate. They provide no further reasoning behind choosing this value. To investigate their finding in a broader range, we experiment by increasing the learn rate of new layers by $\text{NewLR} \times \text{GlobalLR}$, where $\text{GlobalLR} = 0.005$ i.e. global learn rate and $\text{NewLR} \in \{5, 10, 15, 40\}$. For evaluating these experiments with incremental learn rate at every iteration, we follow the same fine-tuning scheme. Our experiments tend to relate the increment of learn rate to the size of the appended layer and similarity between source and target sets as shown in Figure 7. More precisely, appending 512, 1024, 2048 and 4096 neurons tend to perform better when their learn rates are increased in the range of (5-10), (10-20), (15-25) and (25-40) times global rate respectively. This can be attributed to the fact that more FC neurons introduced in module need greater learn rate to keep pace with the pre-existing units. Our empirical results also demonstrate that new layers introduced in module for target sets which have more semantic label similarity with source set desire for comparatively less increment of learn rate than distant target sets. Therefore, we claim there is no optimal increment for all combinations rather it depends on target set and size of appended layer.

5. Conclusion

In this paper, we propose a depth augmented network for transfer learning that increases its depth capacity beyond pre-trained classification layer subsequent to demonstrating that this layer is efficient for fine-tuning. We provide in-depth analysis to prove our hypothesis and demonstrate by thorough experimentation that classification layer is vital to proposed network for transfer learning. To evaluate proposed approach, we design an incremental fine-tuning scheme to execute diagnostic layer wise fine-tuning. Our evaluation of proposed approach on different benchmark datasets shows that growing layers beyond classification layer outperforms existing transfer learning approaches. Moreover, we observe from diagnostic analysis that depth augmentation requires normalization followed by scaling scheme to tune pre-trained and new parameters at a steady speed to encourage better transfer learning. Our incremental fine-tuning scheme portrays an interesting outcome that fine-tuning except highly generic layers perform best in target tasks. In addition we notice that increasing network depth by appending multiple layers facilitate network that is compatible with improved target task performance.
to learn even better by increasing representational capacity. Finally, we find that newly appended layers desire for greater learn rate than existing layers. Our empirical analysis show learn rate of transferred layers depend on target dataset and number of FC neurons introduced in the layer itself.

References

[1] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 36–45, 2015. 1, 2, 4, 5, 6, 7
[2] N. Becherer, J. Pecarina, S. Nykl, and K. Hopkinson. Improving optimization of convolutional neural networks through parameter fine-tuning. Neural Computing and Applications, pages 1–11, 2017. 2
[3] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT’2010, pages 177–186. Springer, 2010. 3
[4] M. L. Collins. Handbook of developmental cognitive neuroscience. MIT press, 2008. 2
[5] C. Cortes and V. Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995. 1
[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009. 1
[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning, pages 647–655, 2014. 1
[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 4
[9] K. Fukushima and S. Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern recognition, 15(6):455–469, 1982. 1
[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014. 1, 2
[11] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007. 4
[12] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 447–456, 2015. 1
[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 1
[14] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006. 1
[15] M. Huh, P. Agrawal, and A. A. Efros. What makes imagenet good for transfer learning? arXiv preprint arXiv:1608.08614, 2016. 1
[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML’15, pages 448–456. JMLR.org, 2015. 3
[17] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilevsk. Learning visual features from large weakly supervised data. In European Conference on Computer Vision, pages 67–84. Springer, 2016. 6
[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012. 1, 5
[19] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. arXiv preprint arXiv:1506.04579, 2015. 3
[20] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605, 2008. 7
[21] R. Mormont, P. Geurts, and R. Marée. Comparison of deep transfer learning strategies for digital pathology. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 2262–2271, 2018. 1
[22] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10), pages 807–814, 2010. 3
[23] M.-E. Nilsson and A. Zisserman. Automated flower classification over a large number of classes. In 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, pages 722–729. IEEE, 2008. 4
[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1717–1724, 2014. 1
[25] S. J. Pan and Q. Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2010. 1
[26] M. Pickett, R. Al-Rfou, L. Shao, and C. Tar. A growing long-term episodic & semantic memory. arXiv preprint arXiv:1610.06402, 2016. 2
[27] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3716–3724, 2015. 6
[28] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. arXiv preprint arXiv:1606.04671, 2016. 1
[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1

[30] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 1, 6

[31] O. Sigaud and A. Droniou. Towards deep developmental learning. *IEEE Transactions on Cognitive and Developmental Systems*, 8(2):99–114, 2016. 2

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 5

[33] D. Strigl, K. Kofler, and S. Podlipnig. Performance and scalability of gpu-based convolutional neural networks. In *18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2010)*, pages 317–324. IEEE, 2010. 1

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[35] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016. 1, 2

[36] Y. Tamaazousti, H. L. Borgne, C. Hudelot, M. E. A. Seddik, and M. Tamaazousti. Learning more universal representations for transfer-learning. *arXiv preprint arXiv:1712.09708*, 2017. 1, 6

[37] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor. A deep hierarchical approach to lifelong learning in minecraft. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 2

[38] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, pages 640–646, 1996. 2

[39] S. Thrun and J. OSullivan. Clustering learning tasks and the selective cross-task transfer of knowledge. In *Learning to learn*, pages 235–257. Springer, 1998. 2

[40] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 4

[41] Y.-X. Wang, D. Ramanan, and M. Hebert. Growing a brain: Fine-tuning by increasing model capacity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2471–2480, 2017. 2, 3, 6, 8

[42] S. Yang and D. Ramanan. Multi-scale recognition with dagnns. In *Proceedings of the IEEE international conference on computer vision*, pages 1215–1223, 2015. 1

[43] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 1, 2, 7

[44] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1