university of groningen

faculty of science and engineering

artificial intelligence

Deep Learning

Assignment 1

Group 12

Aydan Yagublu (s3980804), Eric Brouwer (s3934640),
Lauren Kersten (s3950905), Thijs van der Laan (s3986721)

Research Paper

Do Deep Convolutional Nets Really Need to be Deep (Or Even Convolutional)?

I. High-Level Description / Summary

The paper "Do Deep Convolutional Nets Really Need to be Deep (or Even Convolutional)?" by Urban et al. (2016) discusses much theoretical work that exists concerning the representative capacity of artificial neural networks. So far, empirical research has suggested difficulties can arise when training shallow networks to be as accurate as deep ones. Research in computer vision suggests a preference for deeper networks, for example, when training on SIFT-features (Dauphin and Bengio, 2013) or when considering deep convolutional networks under a parameter budget (Simonyan and Zisserman, 2014). This preference for deeper networks over shallow ones exceeds other fields as well, such as speech acoustic modelling (Seide et al., 2011).

However, research by Ba and Caruana (2014) demonstrates another phenomenon. By first training state-of-the-art deep models and then training shallow student models to mimic them, they demonstrate that with a fixed number of parameters, these shallow nets are sometimes able to learn the same functions as deep nets. However, they note that it is not yet understood why this is possible and how it works. Whilst these results are interesting, Ba and Caruana (2014) mainly demonstrate this equal performance on the TIMIT dataset as a speech recognition benchmark, where convolution is less important than more computer-vision-oriented tasks such as classification with the CIFAR-10 and ImageNet datasets, amongst others. Consequently, their deep models for the TIMIT dataset only use one convolutional layer.

Ba and Caruana (2014) also present research for the previously mentioned CIFAR-10 dataset, though these results were less convincing in comparison to those from the TIMIT speech recognition modelling. To obtain accurate shallow models on the CIFAR-10 benchmark, they had to add at least one convolutional layer and include additional parameters until the number of parameters in the shallow models was about 30 times larger than that of the deep models. After these modifications, it still scored below the teacher model, whose performance was not considered to be state-of-the-art.

Therefore, Urban et al. (2016) attempted to reproduce and elaborate on the findings of Ba and Caruana (2014) whilst focusing more on the convolutional side of the neural networks. Their methodology is kept as similar to the one in Ba and Caruana (2014) as possible to rule out the effect of confounding variables on their findings.

While the goal is to build on Ba and Caruana (2014)'s findings and also use the CIFAR-10 dataset, there are some differences within the methodologies of both papers. In the work of Ba and Caruana (2014), the deep models were not state-of-the-art, and the number of parameters for the shallow models was not kept fixed to match that of the deeper models. Both these things have been done in the work of Urban et al. (2016), where their deep models actually reached state-of-the-art accuracy and a parameter budget was set.

A. Methodology

The methodology of the current paper consists of a number of applied techniques. The methods considered are model compression (distillation), data augmentation, Bayesian hyperparameter optimization, and training shallow models to mimic an ensemble of the Deep Convolutional CIFAR-10 models, amongst others.

Distillation: Model compression, in essence, consists of training a compact model to approximate the function learned by another larger and more complex model. It works as follows: unlabeled data is passed through the teacher model, which is larger and highly accurate, in order to obtain the scores it predicts. Then, instead of training the student model on the original dataset, this student model is trained to mimic these predicted scores. The methodology from Bucila et al. (2006) that applied this technique was improved on by Hinton et al. (2015) by adding a parameter that allows specifying the trade-off in importance between the soft-targets, obtained from the teacher model and the hard-targets as provided by the original training data. Urban et al. (2016) made use of this technique as well.

Generally, the knowledge passed on from the teacher model to the student takes the form of relative probabilities. Therefore, this so-called *dark knowledge* contains the probabilities of all classes instead of just giving the correct class. Interestingly, this distillation can allow shallower models to achieve close to similar accuracy as their larger, deeper teacher models, which they are trained to mimic. However, it is not yet well understood why this training on *dark knowledge* leads to higher results compared to training on the one-hot target vectors that follow from the original training set.

Mimic Learning (Logits): In this paper, the soft targets used for mimic learning come from an ensemble of deep nets trained on the original training data from CIFAR-10. These deep models are trained with a softmax output and cross-entropy loss function.

The student models, however, are trained with cross-entropy on the ten log probability values before the softmax activation. These logits of predicted probabilities are the earlier mentioned *dark knowledge* that help in learning the relationships across all outputs that the deep teacher models have learned.

Linear Bottleneck: Additionally, following the methodology of Ba and Caruana (2014), it was found that the wide layers that come with shallower nets become very large when keeping the number of parameters similar. Subsequently, these extremely wide layers lead to slow training when using backpropagation. Therefore a linear bottleneck layer was introduced between the input and non-linear layers to improve the learning speed.

Urban et al. (2016) incorporated these linear bottleneck layers in their student networks with 0 or 1 convolutional layers but did not find them necessary when the number of convolutional layers exceeded 1.

Bayesian Hyperparameter Optimization: For hyperparameter optimization, in all experiments, Bayesian Hyperparameter Optimization with Gaussian Process Regression is employed in order to explore all hyperparameters that govern learning. To achieve this task, they specifically utilize Spearmint (Snoek et al., 2012) in their implementation.

Training Data & Augmentation: For this study, only the CIFAR-10 dataset was utilized; however, a series of preprocessing steps and data augmentation techniques were also employed. The CIFAR-10 dataset contains 80 million images; however, for this study, a labelled subset of 60,000 32x32 pixel natural images from 10 object classes was selected. The dataset is split into 50,000 training images and 10,000 as a validation set for Bayesian optimization. The images were preprocessed by subtracting the mean and dividing by the standard deviation of each image vector. The study used 40,000 training images and validated on the remaining 10,000 images. For data augmentation, they employed the HSV data augmentation technique by shifting hue, saturation, and value by random values, with saturation and value values, scaled globally. Moreover, the previously mentioned constants (e.g. hue, saturation, and value) were held as additional hyperparameters during the model optimization process. Additionally, all training images have a 0.5 probability of being randomly mirrored left-right. They are also further scaled and jittered by cropping windows of size 24×24 up to 32×32 at random locations and then scaling them back to 32×32.

Moreover, all these augmentations are generated through online augmentation, meaning that they are computed on the fly. However, for the student models, 160 epochs worth of randomly augmented training data are pre-generated and evaluated on the ensemble's predictions. From there, all the data and predictions, the logits, are saved on a disk to be later utilized by all student models. Therefore all student models train similarly on the same training data, leading to a less computationally expensive training process.

Learning-Rate Schedule: All of the models designed for this study were trained using Stochastic Gradient

Descent with Nesterov Momentum. This momentum value, as well as the initial learning rate of the models, were selected by the previously mentioned Bayesian Hyperparameter Optimization. The learning rate adapts to the performance of the models, using the validation error as a form of reference measurement. The learning rate is halved if the validation error has not decreased for 10 subsequent epochs and must wait for at least another 8 epochs before changing again. Training is concluded if the validation error has not decreased for 30 consecutive epochs or if the learning rate has decreased by a factor of more than 2000. The authors mention that even though this is not the ideal method of optimization for each model individually, it provides a fair training session for each model in a shorter amount of time than otherwise possible.

Super Teacher: The 'super teacher' in this research consists of an ensemble of 16 deep convolutional CIFAR-10 models. As the teacher model in Ba and Caruana (2014), research was not state-of-the-art as they did not use data augmentation, and the ensemble only had an accuracy of 89% with their deepest model only containing 3 convolutional layers; the goal of the present research was to design a state-of-the-art model with the highest accuracy possible to compare the shallow models too.

In total, 129 deep convolutional models were trained with spearmint, of which 18 hyperparameters were optimized: initial learning rate, momentum, l_2-weight-decay, initialization for scaled initial weight values, four dropout rates, 4 constants used to scale the layer width of the network and 5 constants used for the HSV data augmentation.

As it is possible to create an even more accurate model by combining top-performing deep CNNs, the authors decided to create an ensemble of 16 convolutional nets they designed which reached very impressive performance compared to previously published works (94% for the validation set and 93.8% for the test set). The ensemble then is used to re-label the data by taking the average of the logits from each of the models within the ensemble, which is then used to train the shallow student models

Training Shallow to Mimic Ensemble Deep CNN: The student models were trained with 1, 3.16, 10, and 31.6 million trainable parameters on the precomputed augmented training data. The data was re-labeled by an ensemble, and the student models utilized various configurations, such as shallow fully-connected nets and CNNs with either 1, 2, 3, or 4 convolutional layers. These models were designed with fixed numbers of trainable parameters, controlled by a scale factor (between 0 and 1) assigned to each hidden layer which would control the width of the layer it is assigned to. From there one of these factors will be left out while the values of the others are taken. The left-out factor is then computed to match the number of trainable parameters for the model.

The hyperparameters optimized during the training included the initial learning rate, momentum, scaling of learnable parameters, and scaling of input pixel values. Weight decay and dropout were intentionally not optimized or utilized during training, as they were found to have a negative impact on student models with up to 40 million parameters.

B. Results

The paper presents the findings of deep convolutional model hyperparameter optimization using Bayesian optimization paired with weight decay on the CIFAR-10 dataset. As more convolutional layers and parameters are added, the models' accuracy increases as well, according to the data. The accuracy of student models with various numbers of convolutional layers is also compared to the accuracy of the original models in the paper. Distillation training was used to train the student models, which were discovered to be more accurate than those trained on the initial hard 0/1 (one-hot-encoding) labels as seen from the compression gap line in Figure 1. The paper further compares the student models trained in this paper to the student model developed by Ba and Caruana (2014), which was less accurate and larger.

The best three models' architecture, parameter count, and accuracy were discussed in the authors' presentation of the deep convolutional model results. This also includes a group of 16 CNNs and the accuracy of the teacher models. These results demonstrated that as the number of convolutional layers increases, so does the model's accuracy. The best-performing deep model had an accuracy of 91.75%, while the ensemble of 16 CNNs had a maximum accuracy of 93.8%.

The student models were trained using distillation and were discovered to be more accurate than their counterparts trained on the original hard targets. Student models with different numbers of convolutional layers were compared to the accuracy of the original models. This is illustrated in Figure 1, showing the average performance of each of the model parameter setups. The best-performing student models contained 3-4 convolutional layers and 10-million to 31-million

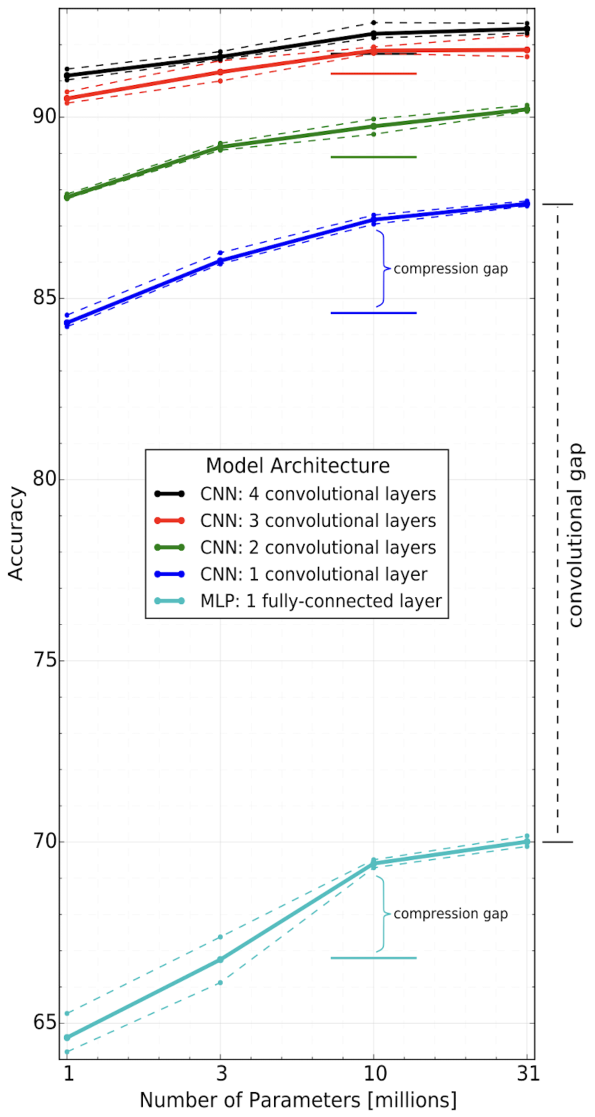

Fig. 1: Accuracy of Shallow Student Networks Trained to Mimic Ensemble of Deep CIFAR-10 Models.

parameters. Furthermore, the student model trained by Ba and Caruana (2014), which was less accurate and larger than the student models trained in this paper, was also compared to the accuracy of the student models.

While not showing similar accuracies for the compressed student models compared to their teacher models, the results show clearly that distillation and hyperparameter tuning are viable methods for training precise deep convolutional networks on CIFAR-10. The research offers insightful information on the relationship between the model accuracy and the number of convolutional layers and parameters. The comparison to the student model trained by Ba and Caruana (2014) also demonstrates the developments in deep learning since they created their model.

C. Discussion and Conclusion

Despite not being able to design a shallow model that is capable of comparable performance to that of the state-of-the-art deep CNN ensemble in the paper, the models they produced for this study are some of the highest accuracy models to have been published regarding the CIFAR-10 dataset.

When comparing against Ba and Caruana (2014) whose parent ensemble of 4 CNN models (all with 3 convolutional layers) only reached an accuracy of 89%, the student models designed with distillation here outperform the previously mentioned parent ensemble with an accuracy of 90% given 2 convolutional layers and 92% with 3 layers. This improved student performance still does not compare to that of the state-of-the-art ensemble model's performance from the current paper (93.8%). However, the noticeable improvement in performance is largely due to the distillation method using the soft targets during the training of the shallow student models.

The authors make it clear that they did not include the data augmentation hyperparameters in the Bayesian optimization 'for computational reasons' as the results only sometimes improved (and even if only very slightly). They also mention that the optimization process could have been tuned more individually for each model, which could have improved the observed performances even further. They concluded that the inclusion of a dropout layer only hurt the accuracy of the models as it is assumed that the distillation process provides enough regularization for the student models.

Overall, the aim of the study was to see if it was possible to train shallow student convolutional neural networks to be as accurate as state-of-the-art convolutional models with the same number of parameters on the CIFAR-10 dataset for image classification. Though they did not succeed in achieving said similar accuracies in this study, they did, however, find that distillation and model compression help improve the accuracy of shallow models as well as designed some of the top-ranking state-of-the-art models to be published regarding the scope of the problem.

II. The 7 W's

What Is The Problem?

As the title suggests, the paper "Do Deep Convolutional Nets Really Need to be Deep (Or Even Convolutional)?" attempts to answer the question regarding whether shallow feed-forward neural networks need to be deep and convolutional in order to learn the complex tasks and functions previously learned by deep convolutional models (Urban et al., 2016). Previous research by

Ba and Caruana (2014) has shown that when training shallow models to mimic the deep models, shallow models are able to learn functions and yield accuracies comparable to those demonstrated by state-of-the-art deep models. The problem now that this paper aims to tackle is to build on the research of Ba and Caruana (2014) by designing several models with varying levels of convolution, parameters, and architectures to systematically determine whether or not convolutional networks need to be deep.

Why Is It Important?

It is useful to know the importance that depth plays when building network architectures in order to ensure robustness. Since it is generally computationally cheaper to train a shallow network, the study aims to explore whether a shallow network can achieve similar results to a deep network. Given that deep networks are exponentially more expensive in computational power, if similar results can be achieved with shallow networks, it would be an indication that for the given task at hand, there is no reason to utilize deep networks.

Simply put, a shallow model is simpler than a deep model, which allows them to be computationally cheaper, faster, and more interpretable, making them more accessible to those with limited resources.

What Have Other People Done About The Problem?

As stated before, the problem in the current study is to see whether shallow networks must be deep (or even convolutional) in order to succeed while executing complex tasks. In previous research, deeper models have been preferred (He et al. (2016); Eigen et al. (2013)) and have been found to have a better performance than shallow ones (Seide et al. (2011)), but Ba and Caruana (2014) provide evidence that shallow nets are sometimes capable of learning the same functions as deep networks with comparable size.

The results presented by Ba and Caruana (2014) indicate that some shallow models are better at mimicking deep models as opposed to learning directly from the data that was used to train the deep parent model. However, these results have been derived from models that do not rely heavily on convolutional layers (despite having one convolutional layer) since they are utilizing the TIMIT dataset for speech recognition tasks where convolution is not as important as for image recognition tasks such as those using the CIFAR-10 or ImageNet datasets.

Ba and Caruana (2014) also investigated whether shallow models were able to mimic deeper ones using the CIFAR-10 dataset. They found less impressive results than before with the TIMIT dataset; however, they also were able to show that these shallow models with at least one convolutional layer and about thirty times as many parameters as the deep parent networks were able to learn with comparable levels of accuracy to the deep nets, though always slightly less accurate.

Why Is It Not Sufficient?

When utilizing the CIFAR-10 dataset, Ba and Caruana (2014) found that to train accurate shallow neural networks, they had to include at least one convolutional layer in the shallow model, and the number of parameters had to be increased by a factor of 30 compared to the deep models. However, in their findings, Ba and Caruana (2014) do not address the question of whether a shallow network can still achieve similar results when the number of parameters is restricted. Furthermore, their teacher models do not really achieve state-of-the-art accuracy either. As such, Urban et al. (2016) contend that this study is insufficient to address the issue of whether convolutional models require depth in order to achieve state-of-the-art performance.

What Is Proposed To Be Done Differently?

Unlike in the paper by Ba and Caruana (2014), in the assigned paper, Urban et al. (2016) compare the shallow models to state-of-the-art deep convolutional models whilst restricting the number of parameters in the shallow models to be comparable to the number of parameters in the deep convolutional models. Additionally, they aim for their teacher models to actually achieve accuracies that can be considered state-of-the-art. Nevertheless, because they anticipate that their results may slightly differ, they follow their methods as closely as possible to eliminate the possibility that the results differ because of changes to the methodology.

Why Is It Better?

Urban et al. (2016) specifically go on to compare the shallow models to state-of-the-art deep convolutional models. The term state-of-the-art is used to refer to the current best-performing models in a given field, thus by comparing shallow models to these state-of-the-art models, the researchers are able to better understand the importance of having a "deep" model in regard to high performance. Moreover, by doing so, the comparisons are more meaningful, and their research is more up-to-date and relevant. If the authors had utilized older models, the comparison between them and their shallow models could give false impressions about the potential of shallow models compared to deeper ones.

Additionally, by restricting the number of parameters in the shallow models to be comparable to those in the deep convolutional models, Urban et al. (2016) ensure a more accurate assessment of the trade-offs between model complexity and performance. Through this restriction, they are able to evaluate whether the difference in performance between shallow and deep models is solely due to the depth of the model. By doing so, their research contributes to the broader understanding of the capabilities and limitations of different types of network models.

What Is Left To Be Done?

The paper concludes with the researchers stating what further research must be done. They write: 'The question remains why medium-depth convolutional models trained with distillation are more accurate than models of the same architecture trained directly on the original training set' (Urban et al., 2016). This follows their findings of increased performance when their models, designed to mimic their state-of-the-art deep parent model, outperformed similarly structured models trained on the dataset used to train the said deep parent model.

Furthermore, the authors also mention that if they are able to find a shallow net that performs as well as their deep parent net, then it means there are no instances where convolutional nets must really be deep. However, it also states that if they are unable to produce a shallow net that performs as well, or even better than their deep parent net, it does not necessarily imply that there does not exist such a shallow model capable of this but rather they could have been unable to produce one with such performance in the present study. This means that with more research and modelling, perhaps a shallow model with comparable, or even improved, performance could be found.

How Is The Paper Related To Deep Learning? (as a field)

The discussed paper is highly related to deep learning as it addresses one of the field's fundamental questions, namely, whether a network truly has to be 'deep' in order to achieve state-of-the-art performance. It attempts to provide an answer to whether shallow networks with a comparable number of parameters can achieve similar performance to networks that utilize deep learning.

If the researchers were to find evidence that suggested that shallow models can achieve comparable results to deep networks with only a few convolution layers and a restricted number of parameters, they would challenge the upheld notion in the field of deep learning, which denotes that deep models are more powerful and accurate than shallow ones. Their research would demonstrate that simply adding more convolutional layers to a network does not result in a model yielding better performance.

Additionally, the study highlights that in a previous study by Ba and Caruana (2014), researchers had found equal performance between deep and shallow convolution nets when experimenting on the TIMIT speech recognition benchmark, but when extending their experiments to the CIFAR-10 dataset, the results were less promising. Urban et al. (2016) build upon these findings by further exploring the CIFAR-10 dataset, wherein they conclude that when restricted to the same number of parameters, deep convolutional nets learn models for CIFAR-10 that are significantly more accurate than shallow convolutional models. These findings emphasize the importance of considering task-specific characteristics when designing deep learning models. The results on the TIMIT speech recognition benchmark are better than those of the CIFAR-10 dataset, as convolution is less important for speech recognition than other problems, such as computer-vision-oriented tasks. They demonstrate that there is not a one-size-fits-all approach to deep learning and that further research is required to determine which tasks benefit the most from which network architectures.

References

Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? In *NIPS*.

Bucila, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Knowledge Discovery and Data Mining*.

Dauphin, Y. N. and Bengio, Y. (2013). Big neural networks waste capacity.

Eigen, D., Rolfe, J., Fergus, R., and LeCun, Y. (2013). Understanding deep architectures using a recursive convolutional network. *arXiv preprint arXiv:1312.1847*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network.

Seide, F., Li, G., and Yu, D. (2011). Conversational speech transcription using context-dependent deep neural networks. In *Twelfth annual conference of the international speech communication association*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Urban, G., Geras, K., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., Mohamed, A., Philipose, M., and Richardson, M. (2016). Do deep convolutional nets really need to be deep (or even convolutional)?