A knowledge distillation approach to increase the accuracy of lightweight convolutional neural networks

Instructor

Prof. Dr. M. Vojnovic

Teaching Assistant

T. Xu

Candidate number

47602

December 6, 2020

Abstract

Predicting data using complex models is cumbersome and can be too computationally expensive to allow deployment to many users, especially if the complex models are large neural networks. This paper aims to achieve high accuracy on image classification tasks using small and fast-to-execute models. These models do not in themselves achieve high accuracy, but rather require the more complex models to train them. This is achieved using a combination of knowledge distillation, a technique that distills knowledge from the output of a complex teacher network to a small student network, and hint layers, a technique that distills knowledge from the output of an intermediate layer from a complex teacher network in a similar fashion. The performance gains are measurable but not considerable.

1 Introduction

Operating large models under constrained computational training or inference budgets remains challenging. The best-performing state-of-the-art Convolutional Neural Networks (CNN) are nearly impossible to train on commodity hardware. A consequence of this is that running these CNN in real-time, such as in the setting of self-driving cars, is unfeasible. More lightweight CNN are necessary for such settings. Such networks, however, usually do not attain the accuracy these settings ask for. Furthermore, advances in image classification have relied on increasingly deeper architectures. However, many papers emphasise that very deep neural networks are often over-parameterised to aid generalization [1]. This also calls for improving the performance of more lightweight CNN.

To improve upon accuracy reached by lightweight CNN, a technique called knowledge distillation [4] can be used. This involves training a more complex model, or ensemble of models, referred to as the teacher model, and using their soft outputs as training labels for the more lightweight network, the student model. In this paper knowledge distillation as proposed by Hinton et al. [4] and hint and guided layers as proposed by Romero et al. [7] with feature adaptation as proposed by Chen et al. [2] are combined to train lighter versions of simple CNN in a TensorFlow 2.x framework. When the gap between student and teacher is large, student performance can deteriorate, and therefore a teacher-assistant approach as proposed by Mirzadeh et al. [6] is implemented too. These implementations are carried out in the form of three experiments.

The first experiment is to combine knowledge distillation and hint layers with an adaptation layer in order to attain an improvement in performance for a student model. The teacher and student models are both derived from the VGG architecture. On the CIFAR-10 dataset an improvement of 1.7% in test accuracy using knowledge distillation is attained, and an improvement of 2.5% using both knowledge distillation and hint layers. For CIFAR-100 the improvements are respectively 7.2 % and

10.2%. The second experiment performs multi-step knowledge distillation as in [6], but no gains on test accuracy are possible for the small student model totalling just 130,000 parameters. The third and final experiment tried to attain test accuracy improvements for dense neural networks for image classification tasks, using the VGG teacher network as in the other experiments, but again no improvements are made again.

In the next section the theoretical framework of this paper is described, briefly explaining the relevant theoretical underpinnings of the analyses in this paper. In the third section the method and data employed in the analyses are detailed, and in section 4 the numerical results are presented and evaluated. In the final section the results, their limitations and comparisons to previous literature are presented.

2 Theoretical framework

This section first briefly describes CNN and issues with their interpretability and depth, after which the concepts underpinning the analysis in this paper are explained. These concepts are knowledge distillation, hint layers and teacher assistants.

2.1 Convolutional Neural Networks

CNN were initially built to model the visual cortex in code. Their application is therefore also mostly in image classification and object detection. Although the link between the visual cortex of the brain and CNN is not entirely obvious, CNN are almost universally used in computer vision applications since 2012. The benefits of such models are that the patterns they learn are translation invariant, i.e. these patterns can be anywhere on the image processed, which is not the case for other types of deep neural networks. Furthermore, they can learn spatial hierarchies of patterns, i.e. a first layer will learn small local patterns such as edges, a second layer will learn patterns made out of features of the first layer and so on.

2.2 Interpretability of Neural Networks

The performance of neural networks is much easier to explain in terms of their code than in terms of their properties. For many years researchers from a variety of backgrounds have worked on the problem of compressing large neural networks into simpler and more describable models or systems, to little avail. For instance, for models trained for ImageNet, classification cannot easily be compressed to fewer than about 100.000 parameters. Such large numbers will never be interpretable for a human. For one of the easiest classification problems, classifying the Modified National Institute of Standards and Technology database (MNIST) dataset, a dataset containing handwritten digits, a model performing well cannot readily be compressed in human-readable format [3].

Interpretability is, however, essential. It is demanded for cases involving ethics or in the context of recent EU regulations (General Data Protection Regulation), or more concretely: interpretability is essential for decision-making based on facts instead of seemingly random parameters and numbers. There is a distinction to be made between post-hoc and ad-hoc interpretability, i.e. explaining existing models and constructing new models [5]. Knowledge distillation is an example of improving post-hoc interpretability, as more complex models help in training models that are easier to interpret. This does not make CNN at once entirely understandable, but aids in making them easier to understand.

2.3 Knowledge Distillation

In [1] it was initially proposed to compress deep networks into shallower but wider ones, where the compressed model mimics the logits of the deeper network. This idea was built upon by Hinton et al. [4], where instead of L2 loss (that minimises the error of all the squared differences between the true values and the predicted values), temperature cross entropy loss was proposed to be used.

This entails transferring the generalisation ability of the more complex model to a small model by using the class probabilities produced by the complex model as "soft targets" for training the small model. Usually the models are trained on targets that are encoded as vectors containing only zeros and ones. When the soft targets that the small model is trained on have high entropy, they can provide relevant information for the small model, as the "soft targets" could indicate to the small model that some of the images to be classified resemble each other, for instance the numbers 2 and 7 in the MNIST dataset. Furthermore, these soft targets contain much less variance in the gradient between training cases, and therefore the small model can often be trained on much less data than the original complex model. Hinton et al. [4] raises the temperature of the final softmax layer in the complex model until its output is suitably 'soft'. Raising the temperature is equivalent to using a softmax output layer with an included T term, as such

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp\left(z_j/T\right)}$$

where z_i are the logits for class i. In training the small model, the weighted average of two different objective functions is used. The first objective function is the cross entropy of the output of the small model's softmax layer having the same temperature as the soft targets computed from the complex model. The second objective function is the cross entropy with the correct labels. The output logits from the final layer of the small model now have a temperature of one. The loss function is thus:

$$\Lambda_{KD} = (1-\alpha)H(y_{true}, P_s) + \alpha H(P_T^\tau, P_S^\tau)$$

where H indicates cross entropy, α the tunable parameter to balance the losses, P_s

the output of the student model, and P_T^τ, P_S^τ the output of the teacher and student model for some temperature τ.

2.4 Hint Layers

Hint learning was initially proposed by [7]. Instead of just using the output to transfer knowledge from a more complex model to a simpler model, the intermediate representation of the teacher is used as a hint to help the training process and improve the final performance of the student. Two intermediate layers from respectively the teacher and student model are chosen. The layers of the student model until the so-called guided layer of this model are trained using the L2 distance between the feature vectors V and Z as a loss function, where Z represents the output of the hint layer, and the V output of guided layer in student network. For the corresponding layers, the number of neurons need to be equivalent. Romero et al. [7] uses a regressor or convolutional regressor to achieve this. Chen et al. [2], however, conclude that using an adaptation layer performs better in accomplishing effective knowledge transfer. The adaptation layer after the guided layer has an output size that is equivalent to that of the hint layer. A 1×1 convolutional layer is used for this to save memory. The height and width of the hint and guided layer must already be equivalent, the adaptation layer simply ensures that the number of channels is equivalent too. The loss function between the guided and hint layer looks as follows:

$$\Lambda_{Hint}(W_{guided}, W_r) = ||u_h(x; W_{hint} - r(v_g(x; W_{guided}); W_r)||^2$$

where u_h and v_g are the teacher and student's deep functions up to their respective hint and guided layer with parameters W_{hint} and W_{Guided}, and r the regressor, or in this case the adaptation layer with parameters W_r.

2.5 Teacher Assistant

When the gap between the student and teacher layer is too large, knowledge distillation can become ineffective because the student model lacks parameters to properly mimic the teacher model. To tackle this, [6] introduced multi-step knowledge distillation, which employs an intermediate-sized network, referred to as the teacher assistant, to bridge this gap. Adding an intermediate teacher assistant indeed improves accuracy, though the improvement is minor.

3 Experiments

This section describes the experiments performed in this study and the data employed in these experiments. The code for all experiments is found in Jupyter Notebook `1.KD10CIFAR.ipynb` and `2.KD100CIFAR.ipynb`. Firstly the CNN employed in all experiments are described.

The complex, or teacher, model is largely based on the Visual Geometry Group (VGG) architecture. This architecture won the ImageNet competition in 2014. The VGG architecture is mostly characterised by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by 2×2 max pooling. Some dense layers are connected to the final max-pooling operation and are then followed by a softmax classifier. The teacher model has 11 convolutional layers, 4 max-pooling layers and 3 dense layers, totalling around 12 million trainable parameters.

The small, or student, model is a simple CNN with two convolution-max-pooling blocks, and one dense layer, totalling around 250 thousand trainable parameters, around 2% of the teacher model's. It must be noted that none of the four experiments in this paper have the intention to outperform other models on the data, and therefore no data augmentation is used for instance. These experiments simply showcase the potential for improvement of student models.

3.1 Hint layers and Knowledge Distillation

The first experiment considers training the teacher model until convergence. Afterwards, similar to [7], the guided layer of the student model is trained using the loss as defined in section **2.4**. Then the student model is trained with a loss function as defined in **2.3**. Such an approach is very similar to that of [7], however, in this paper hint layers are only employed to a specific type of situation where the student model is deeper and thinner than the teacher model, which is less deep but much wider. Combining these two approaches should lead to an increase in accuracy of the student model. In the context of object detection this has been shown to be the case in [2].

3.2 Multi-step Knowledge Distillation

This experiment involves two-stage learning similar to [6]. Instead of using only soft targets to transfer knowledge between the models, however, the teacher now includes a hint layer, and the teacher assistant includes a layer which is simultaneously a guided and hint layer, and the student only a guided layer. This might improve the accuracy gains as accomplished by [6] further. The teacher-assistant in this experiment is the student from section **3.1** and the student is a model consisting of one convolutional, max-pooling and dense layer, totalling 130k parameters.

3.3 From convolutional to dense neural networks

In difficult classification problems, a neural network with only densely connected layers never achieves an interesting accuracy. A dense neural network does, however, have a final dense and softmax layer, precisely as that of the teacher model. In this experiment I intend to find out whether knowledge can also be transferred from a teacher network to a student network not directly suitable for an image classification task.

3.4 Data

To evaluate the performance of the networks in the experiments described above, four datasets were considered. All four are benchmark datasets as this is rather a theoretical than applied study. In addition, in the papers on which these experiments are based, similar datasets were used to evaluate performance. These four datasets are CIFAR-10, CIFAR-100, MNIST and fashion-MNIST. The latter two datasets are rejected after some initial analyses, as the performance gap between student and teacher was negligible and performance gains were arbitrary as, at times, the student would outperform the teacher in terms of test accuracy. The CIFAR-10 consists of 60.000 32×32 colour images in 10 classes, with 6.000 images per class. There are 50.000 training images and 10.000 test images. The classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. These classes are completely mutually exclusive. The CIFAR-100 dataset is the same as the CIFAR-10 dataset, except for the fact that it contains 100 classes instead of 10.

4 Results

This sections describes the results of the experiments. First, performance on both datasets is discussed, and then the additional results are described.

4.1 CIFAR-10

For CIFAR-10, the teacher model achieved a base accuracy of 83.45% (an average of four models with the same architecture). A stochastic gradient descent (SGD) optimiser with Nesterov's momentum was implemented as an Adam optimiser did not converge. The student model achieved a base accuracy of 69.90% (an average of four models with the same architecture). Table 1 outlines the results of the student model for different α and T. α refers to the balancing of the losses in the objective function as defined in section **2.3**. A higher α implies a higher weight assigned to the soft targets, while a lower α implies a higher weight assigned to the true hard targets. T refers to the temperature, as also defined in section **2.3**.

		$\boldsymbol{\alpha}$				
		0.3	0.5	0.7	0.9	
T	2	69.71%	69.66%	70.80%	69.66%	69.95%
	4	69.56%	69.51%	69.81%	69.86%	69.68%
	6	70.13%	70.32%	70.61%	69.60%	70.16%
	8	70.53%	68.97%	70.08%	68.39%	69.49%
	10	70.47%	69.17%	70.27%	70.20%	70.02%
	12	70.11%	**71.10%**	69.92%	71.09%	70.55%
	14	70.07%	69.97%	69.00%	69.09%	69.35%
		70.08%	69.81%	70.07%	69.69%	

Table 1: Accuracy on CIFAR-10 for different α and T

The table above indicates that no particular α performs best, and that there is no clear relation between α and test accuracy. Hinton et al. [4] indicates that a higher α would perform best, as soft targets are more insightful than hard targets, but these results seem to contradict this. Furthermore, the relation between temperature T and test accuracy is not abundantly clear either. Although it seems to be the case that an increasing T increases test accuracy, and then plateaus again after a particular T, in this case 12. This is in line with [4], where it was claimed that a 'suitably' soft set of targets ought to be found. Some averages of rows and columns in fact are outperformed by a model trained without knowledge distillation. The table below demonstrates the results of implementing a hint layer on top of the best performing KD model ($\alpha = 0.5, T = 12$).

Algorithm	**# parameters**	**Base**	**KD**	**KD+HT**
Student	∼0.25M	69.90%	71.10%	71.72%
Teacher	∼12M	83.45%		

Table 2: Final accuracies on CIFAR-10

The performance gain is small, but implementing a hint layer consistently improved test accuracy. The final gain is 2.5%. This gain is much smaller than that in [7], which is understandable as these gains were measured on student models that were much deeper (but thinner) than their teachers. A general consensus is also that deeper nets outperform less deep nets, despite their smaller number of parameters.

4.2 CIFAR-100

For CIFAR-100, the teacher model achieved a base accuracy of 50.15% (an average of four models with the same architecture). An SGD optimiser with Nesterov's momentum was implemented as an Adam optimiser did not converge. The student model achieved a base accuracy of 35.47% (average of four models with the same architecture). Table 3 outlines the results of the student model for different α and T, with α and T as described in the CIFAR-10 section.

		$\boldsymbol{\alpha}$				
		0.3	0.5	0.7	0.9	
	2	35.05%	35.71%	35.80%	35.90%	35.62%
	4	35.21%	35.90%	35.55%	36.40%	35.76%
	6	36.74%	35.20%	34.78%	36.18%	35.72%
$\boldsymbol{T}$	8	34.36%	37.80%	33.06%	35.26%	35.12%
	10	37.64%	**38.04%**	36.90%	36.33%	37.22%
	12	34.75%	36.26%	36.05%	36.41%	35.87%
		35.62%	36.48%	35.35%	36.08%	

Table 3: Accuracy on CIFAR-100 for different α and T

Again, no particular α outperforms the others consistently and the relation between α and test accuracy is not obvious. The relation between temperature T and test accuracy is similar to that on CIFAR-10, where increasing temperature leads to a higher test accuracy, until a certain plateau, which is $T = 10$ in this case. The table below demonstrates the results of implementing a hint layer on top of the best performing knowledge distillation model ($\alpha = 0.5, T = 10$).

Algorithm	**# parameters**	**Base**	**KD**	**KD+HT**
Student	~0.25M	35.47%	38.04%	39.70%
Teacher	~12M	50.15%		

Table 4: Final accuracies on CIFAR-100

The performance gain is small, but implementing a hint layer again consistently improved test accuracy. The final gain is 10.15%, which is much more than that on CIFAR-10.

4.3 Additional results

The two-stage training experiment as described in section **3.2** yielded unsatisfactory results. The test accuracy of the small student network did not improve, neither from training it on teacher soft targets nor from training it teacher-assistant soft targets, nor from using a hint layer between the teacher-assistant and the student.

The experiment as described in **3.3** also yielded unsatisfactory results. Many different neural networks with only dense layers were trained on both datasets, but the test accuracy consistently fluctuated between two values, for instance 20% and 30%. Therefore no consistent results could be found after implementing knowledge distillation on the final layer of these networks.

5 Discussion

This section first present the conclusion of all experiments, after which is discusses the limitations of the analysis and avenues for future research.

In this paper three experiments were carried out. The first experiment was to combine knowledge distillation and hint layers with an adaptation layer in order to attain an improvement in performance for a student model. The teacher and student models were both derived from the VGG architecture. On the CIFAR-10 dataset an improvement of 1.7% in test accuracy using knowledge distillation was attained, and an improvement of 2.5% using a combination of knowledge distillation and hint layers. For CIFAR-100 the improvements were respectively 7.2 % and 10.2%. The second experiment performed multi-step knowledge distillation, but no gains on test accuracy were possible for the small student model totalling just 130,000 parameters. The third and final experiment tried to attain test accuracy improvements for dense neural networks for image classification tasks, using the VGG teacher networks as in the other experiments, but again no improvements were made.

The results in this paper show limitations of previous literature. In [4], [6] and [7] considerable test accuracy improvements are attained. However, these results mostly hold for the specific use cases in their respective papers. The accuracy improvements from [7] for instance are only proved for student networks that are deeper than their teachers in this paper. Indeed, the hint layers still lead to accuracy improvements in this paper, but these improvements are minor compared to what [7] show. Furthermore, the teacher-assistant approach from [6] did not work in this paper. Again, this is most likely because the results from that paper do not generalise to the setting were the student model is very small (130k parameters) and the teacher-assistant model is also very small (260k parameters). This does not refute previous research, but mostly demonstrates its limitations. Using hint layer for both stages could, however, still lead to an interesting improvement of test accuracy if for instance the teacher-assistant model would be sized between the teacher and student model (some millions of parameters), and this would be interesting to explore for future research. Lastly, using dense neural networks instead of convolutional neural networks is simply unfeasible.

References

[1] Ba, J., Caruana, R. (2014). Do deep nets really need to be deep?. In *Advances in neural information processing systems* (pp. 2654-2662).

[2] Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M. (2017). Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems* (pp. 742-751).

[3] Fan, F., Xiong, J., Wang, G. (2020). On Interpretability of Artificial Neural Networks. *arXiv preprint arXiv:2001.02522.*

[4] Hinton, G., Vinyals, O., Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531.*

[5] Lillicrap, T. P., Kording, K. P. (2019). What does it mean to understand a neural network?. *arXiv preprint arXiv:1907.06374.*

[6] Mirzadeh, S. I., Farajtabar, M., Li, A., Ghasemzadeh, H. (2019). Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393.*

[7] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., Bengio, Y. (2014). Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550.*