Neural Network Compression
The Functional Perspective

Israel F. Mason-Williams
Department of Computer Science
University of Cambridge
ifm24@cam.ac.uk

Abstract

Various compression techniques, such as knowledge distillation, pruning, and quantization, have gained attention due to their ability to reduce the computational costs associated with model inference through model compression. These techniques have paved the way for a new era of on-edge machine learning, enabling the deployment of comprehensive machine learning systems. The efficacy of bespoke compression methods is often evaluated through the proxy of accuracy to judge the resulting model. This study aims to explore the functional divergence between compressed and uncompressed models and investigate how these deviations vary across different compression techniques and architectures as a new way to understand compression methods. The results indicate that quantization and pruning create models that are functionally similar to the original model, with the caveat that the optimal percentage pruned is architecture and dataset-dependent. In contrast, we show that knowledge distillation creates models that do not functionally approximate their teacher models, resembling the dissimilarity of function observed in independently trained models. Therefore, we verify, via a functional understanding, that quantisation and pruning are compression methods, that knowledge distillation is not a compression method but simply a training regulariser and that no "knowledge" is "distilled" from teacher to student.

The source code for this project is available at: https://github.com/IFMW01/L46_ifm24

1 Introduction

The recent growth in the size and complexity of neural network models towards billion parameter models [16] has fueled unprecedented advancements in various domains of deep neural networks such as computer vision [6] and large language models [12]. While these complex models have demonstrated remarkable performance, their deployment in resource-constrained environments remains challenging due to their high computational costs [19]. Knowledge distillation[11], pruning [15], and quantization have emerged as solutions to bridge the gap between computational efficiency and complexity in neural networks.

The current understanding of compression techniques is that compressed models approximate the original function of uncompressed models [11]. This assumption results from an accuracy and loss-based analysis of the compact model. For example, the underpinning notion of knowledge distillation involves transferring the knowledge encapsulated in a sophisticated "teacher" model to a more streamlined "student" model, and this student will approximate the teacher's function as the student can match or improve performance via knowledge distillation[11]. However, functional analysis of independent models shows that models trained on the same dataset with similar accuracy and loss can form very different functional representations [8]. As a result, using accuracy and loss

Preprint. Under review.

alone to gauge functional similarity is unsound as it assumes that a correlation of accuracy infers a symmetric relationship of functional similarity, which is not apparent for similarly-performing independent models. As a result, this paper asks the following questions to examine the functional preservation in compressed models.

1. Does the use of Knowledge Distillation, Quantization or L1 Pruning result in compressed models that are functionally similar to the uncompressed models?

2. Which compression methods, if any, are the most efficacious for function preservation?

3. How does compression and function preservation scale across multiple architectures and datasets?

A more comprehensive understanding of the relationship between compression and functional preservation is gained through answering these questions. The paper shows that:

- Different α values do not impact the degree of functional similarity between a student and a teacher in knowledge distillation.
- Knowledge Distillation cannot be considered a compression method as it results in a student model that is as functionally dissimilar to the teacher as independently trained models.
- Post Hoc Quantization is the best method for functional preservation across all architectures and datasets.
- L1 Pruning is an effective method for functional preservation across datasets and is partially constrained by the architecture and dataset.
- Compression of a base model needs to be viewed from functional proximity rather than just accuracy, as it can be misleading.

2 Background

2.1 Knowledge Distillation

Knowledge distillation, shown in Figure 1, suggests that a student can successfully approximate accurate internal representations provided by a trained model[11]. In knowledge distillation, a unidirectional backpropagation updates the student's weights while preserving the teacher's. Embodying a nuanced strategy distilling knowledge from a proficient teacher to a student [11]. Knowledge distillation leverages a distillation loss function and a *temperature parameter* to refine the alignment between soft student predictions and soft teacher labels. Additionally, the loss computation incorporates an *alpha factor (α)* ranging between one and zero that weights the loss of the student and distillation loss. A low α value (close to zero) emphasises student-centric information, and an α value close to one tends toward a more substantial reliance on knowledge transfer from the teacher model.

Self-distillation is a type of distillation which has proven helpful in improving the accuracy of a student with the same architecture as a teacher [1]. Literature suggests that in Self-Distillation, the teacher guides the model to an accuracy that exceeds the teacher's, by guiding the student to a flatter minima [20].

Ensemble distillation is a viable way to compress the wisdom of the crowd into a single model [5] to achieve equal or "superior test accuracy" [1]. Via leveraging the diversity of knowledge represented in an ensemble through different functional representations [8] into a single representation.

The overarching notion is that in all subsets of knowledge distillation, knowledge is transferred from the teacher model to a small model [11], meaning that functionally approximate models should emerge.

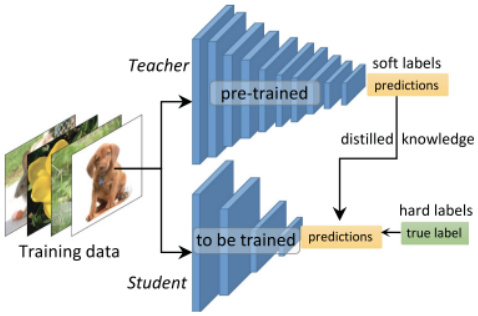


Figure 1: Knowledge Distillation[18]

2.2 Quantization

Quantization is a form of model compression that has gained significance in overcoming bottlenecks associated with deploying machine learning models to edge devices[9]. It is primarily utilised to overcome the challenge of overwhelming memory and computational requirements at the inference time of ever-increasing neural networks. Post-training Quantization is applied as a post hoc method to a conventional model trained with complete 32-bit floating-point precision [4]. The full precision model weights and biases undergo quantization, which involves a transformative shift towards lower-bit precision, ranging from 16 to 4-bit integers. Quantization promises many benefits, such as reduced model size and accelerated inference speed; however, it presents a trade-off as it can result in potential accuracy degradation [9]. To overcome this, fine-tuning a quantised model is possible and allows the model to adapt and compensate for the inherent information loss associated with lower precision representation. Techniques such as Quantised Aware Training strategies come into play to preserve crucial information and minimise the compromise on accuracy at lower bit regimes [3]. In essence, quantization emerges as a compromise between model efficiency and precision, offering a pragmatic solution to the resource-intensive nature of neural while sustaining model accuracies.

2.3 Model Pruning

Model pruning, formally known as Optimal Brain Damage (ODE), follows the notion that network complexity can be reduced by removing parameters with low saliency, referring to those whose elimination minimally impacts the training error[15]. As a general rule, it is observed that parameters with small magnitudes exhibit lower saliency and, therefore, can be removed (pruned) from the overall network [15]. ODE developed into model pruning by employing a magnitude-based weight pruning method that is more computationally efficient and scales better to complex networks [22] such as L1 global pruning. In terms of model compression, structured pruning [2] is the most viable method to accelerate the hardware performances of deep neural networks. Conversely, unstructured pruning does not adhere to a particular geometry or constraint, and consequently, additional information is required to denote sparse locations, which is a bottleneck for efficient computation [2]. Pruned networks can perform with similar accuracy to the original unpruned network, which shows that original models are somewhat over-parameterised.

2.4 Functional Similarity

The study of functional similarity among neural networks offers valuable insights into the diversity of networks and the manifold of functional representations that emerge in models to fit the same datasets. Ensembling, wherein the outputs of models are aggregated to achieve greater performance,[21] shows the ability to leverage a range of diverse functional representations that arise from different random initialisation points to improve accuracy [8]. Exploring functional diversity is crucial to understanding how neural network functions differ and how these characteristics ultimately shape their predictive nature. It is worth noting that models with similar accuracies and losses can exist in entirely different functional spaces[8].

In exploring diversity in ensembles, [8] presents a method for visualising the functional

similarity of any two or more models. Wherein the softmax predicted outputs of a neural network on the test set are reduced to a two-dimensional representation using t-NSE [17]. The t-NSE output can then be plotted to show the functional similarity of two separate neural networks by the distances of the final outputs. The method is effective as it uses the prediction outputs as an abstraction for the function, which allows for spatial comparison.

3 Experimental Set Up

Initial exploration was completed using the LeNet Architecture on the MNIST dataset [7], and these results are also presented in Section 4.1.

To further explore the compression methods' functional similarity, three architectures, LeNet, ResNet-50 [10] and MobileNet [13] Table 1, were trained on CIFAR-10 and CIFAR-100 [14]. The datasets were chosen as they increased in complexity Table 2, allowing the comparison of complexity regimes for each architecture verifying the functional relations of compression methods. The findings shown are evaluated for how compression methods and functional similarity scale with increasing dataset complexity.

Three independent models of the same architecture were trained for each respective architecture and dataset. Each model was initialised, saved and trained for 25 epochs with a batch size of 64, with the final model being saved. After each epoch, the model was evaluated against the test dataset, and the softmax outputs were saved. This allowed a baseline of each model and their functional diversity using the method described in Section 2.4. Following this, each compression method was applied to the base model (B), which is the first independently trained model.

Model	**Trainable Parameters**	**Size (MB)**
LeNet	225,034	0.879
MobileNet V1	3,736,906	14.34
ResNet-50	24,634,980	94.18

Table 1: Architecture Features

Dataset	**Input Dimensions**	**Number of Instances**	**Number of Classes**
MNIST	1 x 28 x 28	70,000	10
CIFAR-10	3 x 32 x 32	60,000	10
CIFAR-100	3 x 32 x 32	60,000	100

Table 2: Dataset Features

3.1 Knowledge Distillation

For all distillation results presented in this paper, a temperature (T) of 3 is kept constant, and the alpha (α) value varies between models $\alpha = [0.1, 0.5, 0.9]$. The alpha values were chosen as it is important to observe if the functional similarity to the teacher is altered depending on how much of the teacher's signal is used during training. Each student model was trained for 25 epochs to match the training provided to the baseline teacher model - with softmax test outputs being saved at each epoch. Self-distillation was explored, giving the student the teacher's initialisation weights and then using the teacher model for distillation. **For each alpha value,the average percentage deviation in accuracy and the standard deviation over three models is reported**, to perform each self-distillation, the random seeds **2,24,42** are used, and the functional landscapes plotted are the random seeds that have the lowest average KL divergence on distillation loss. The appendix presents the functional similarity plots for each architecture and the self-distilled models. An objective of this study is to observe if knowledge distillation produces models that have a functional resemblance to their teacher. Consequently, self-distillation was selected to be tested, given that in a self-distillation scenario, both the teacher and the student have identical learning capacity in terms of parameters, and, therefore, there would be an increased likelihood of functional similarity of the resulting student model to the teacher. Employing self-distillation provides the best opportunity to see the impact of

the distillation process on functional similarity over dependency of reduced capacity, which could result in functions that may be limited in their ability to resemble one another functionally.

3.2 Quantization

Quantization was applied as a post hoc method to the same final model used for the teacher in the distillation process - the softmax prediction values were then saved for the quantised model. The model was quantised at a 16-bit and 8-bit regime to observe if functional differences scaled with bit representations.

3.3 Pruning

L1 Global pruning is applied as a post hoc method on the final model; this allows the functional similarity to be compared when pruning [10%, 30%,50%,70%,90%] of the original network. For this paper, L1 pruning was implemented in Keras without an epoch of fine-tuning. Removing the epoch of fine-tuning was done to allow observation of pruning on functional similarity from an isolated perspective, without allowing the aleatoric nature of training to invalidate any findings.

4 Results

4.1 MNIST

To initially compare the functional similarity of compression methods to a base model, the MNIST dataset with the LeNet architecture was used. Despite LeNet being over-parameterised for MNIST, it provides insight into the impact of compression on function. The base model converges to a test accuracy of 99.15%.

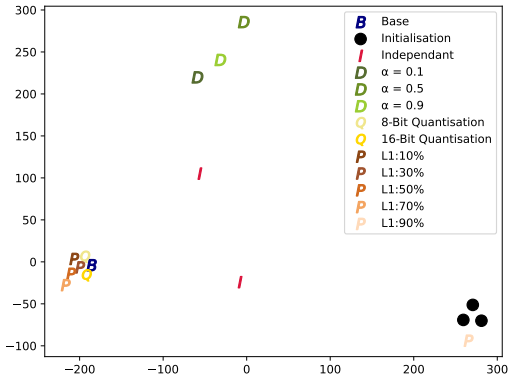


Figure 2: MNIST Functional Landscape for LeNet Model

Architecture	Knowledge Distillation with α			Quantization		L1 Pruning				
	0.1	0.5	0.9	16-bit	8-bit	10%	30%	50%	70%	90%
LeNet	-0.178 (±0.057)	-0.124 (±0.042)	-0.215 (±0.076)	-0.07	-0.05	-0.07	-0.08	-0.01	-1.04	-22.55

Table 3: Percentage deviation in accuracy from the base model on MNIST

Table 3 shows the accuracy deviation from the base model given the application of each compression method. It is evident from the table that each compression method has a minute impact on the model's accuracy - showing that such compression methods can be utilised without largely impacting the original model. Only when the pruning reaches 90% of the network is a dramatic fall-off in accuracy above 2%, suggesting that an optimal prune exists between the boundary of 70% to 90% without significantly impacting the accuracy. However, given that the model is highly over-parameterised for the MNIST dataset, these results are expected.

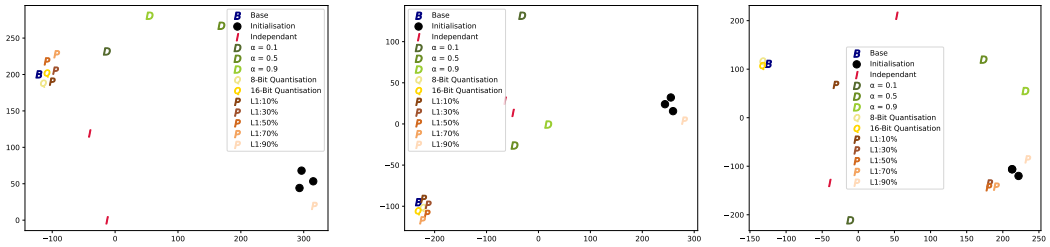
The functional landscape of the base model and compressed models can be seen in Figure 2. To provide perspective, two independent models are also presented. Both independent models have a similar accuracy to the base model within 1-2%. Their presence demonstrates the wide deviation in functional representations between independent models trained on MNIST.

Additionally, Figure 2 shows how post hoc quantization impacts the functional representation of the model. The quantised models remain close to the base model representation, with the 16-bit quantization being the closest representation and the 8-bit quantization being slightly more dissimilar. A similar trend can be witnessed with the L1 pruning regime. As the global pruning factor increases in 20% increments from 10% to 90%, the resulting pruned models move away in functional similarity from the base model. In line with the 22% drop observed in Table 3 at 90% of pruning, the functional representation returns to the initialisation points of the three independent models in Figure 2. A possible explanation could be that the model pruned to such a high degree loses the supporting weights that enabled high test convergence, causing the decrease in accuracy towards initialisation accuracy.

The functional similarity of the self-distillation presents a very different perspective of distillation as a compression mechanism. Given that the knowledge is supposed to be transferred from the teacher model to the student, it would be expected that the student would be close in functional similarity to the teacher with increasing proximity as the α increases. However, the functional landscape shows that for all α values, the distilled models are just as removed from the base model functionally as the independent models, despite the accuracy being less than 1% away from the base model across all α values. Without the proximity in functional representation seen with the quantization and pruning, it is challenging to reconcile distillation as a compression mechanism as it does not create a model that is at all functionally similar to its teacher.

4.2 CIFAR-10

To further validate the findings suggested by the functional representation in MNIST, CIFAR-10 is employed as it is a more complex dataset. More architectures were also introduced to see if the results shown on MNIST are universal across architectures. The base models LeNet, ResNet-50 and MobileNet achieved test accuracies of 63.38%, 67.37% and 68.70 %, respectively.



(a) LeNet (b) ResNet-50 (c) MobileNet

Figure 3: CIFAR-10 Functional Landscape

Architecture	Knowledge Distillation with α			Quantization		L1 Pruning				
	0.1	0.5	0.9	16-bit	8-bit	10%	30%	50%	70%	90%
LeNet	8.74 ($\pm$0.65)	8.93 ($\pm$0.45)	8.57 ($\pm$0.72)	-0.62	-0.33	-0.25	-0.60	-2.95	-14.56	-78.466
ResNet-50	-0.74 ($\pm$1.16)	-3.41 ($\pm$1.35)	0.09 ($\pm$0.9)	0.00	-0.56	-0.08	0.48	-0.03	-0.04	-85.30
MobileNet	-18.25 ($\pm$0.56)	-20.08 ($\pm$0.77)	-21.15 ($\pm$1.48)	0.02	0.07	-16.84	-83.70	-85.43	-83.84	-85.40

Table 4: Percentage deviation in accuracy from the base model on CIFAR-10

The CIFAR-10 percentage deviation results shown in Table 4 provide insights into which compression methods are best for each model. For example, a ResNet-50 can be pruned to 70% with an insignificant impact on accuracy and has no benefit from using a teacher across the range of α values. Whereas, the LeNet benefits from the self-distillation but does not realise benefits from pruning greater than 50%. It is also evident from the data that all models see an equal fluctuation in deviation when considering the quantization methods. Interestingly, the L1 pruning results show that different architectures

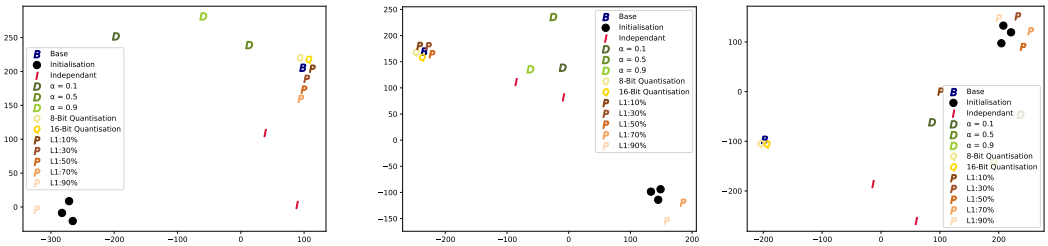
respond uniquely to pruning. This suggests that some architectures have more redundant information held than others, allowing for a harsher pruning regime to be implemented. For example, MobileNet is not resilient to pruning and functionally breaks down at 10% of pruning.

While the efficacy of the compression methods varies between the different architectures for CIFAR-10, the functional landscapes all indicate insights that mirror those witnessed on the LeNet trained on MNIST. The quantised models remain very close to the base model, and likewise for the pruned models of LeNet and ResNet-50. Even on the LeNet, when 70% of the network is pruned, and there is a resultant accuracy decrease of almost 15%, the functional similarity remains very close to the base model. However, when considering ResNet, it displays an apparent insensitivity to pruning, as 70% of pruning results in a marginal loss in accuracy and functional similarity.

Yet again, with knowledge distillation, we see the functional behaviour mimics that of the independent models - highlighting that the teacher does not pass any information that helps the student model approximate its function in any way whatsoever. For the LeNet architecture, where the self-distillation results in higher accuracy, the models appear to be further away from the base model than the independently trained models. The results presented across each model for knowledge distillation further consolidate the notion that it is not a compression mechanism but a different way to train a model, as it does not lead to a functionally similar student.

4.3 CIFAR-100

The final exploration of compression methods on models was completed on CIFAR-100, with the base test accuracy being 33% for LeNet, 37% for ResNet-50 and 29.12% for MobileNet.



(a) LeNet (b) ResNet-50 (c) MobileNet

Figure 4: CIFAR-100 Functional Landscape

Architecture	Knowledge Distillation with α			Quantization		L1 Pruning				
	0.1	0.5	0.9	16-bit	8-bit	10%	30%	50%	70%	90%
LeNet	15.64 ($\pm$1.56)	15.71 ($\pm$0.51)	15.63 ($\pm$1.56)	0.00	-0.16	-0.13	-0.7	-4	-32.31	-85.62
ResNet-50	-5.20 ($\pm$3.89)	0 ($\pm$2.55)	1.56 ($\pm$2.43)	-0.08	-0.16	0.05	-0.16	-1.45	-96.97	-97.34
MobileNet	-26.66 ($\pm$1.17)	-25.63 ($\pm$2.66)	-27.35 ($\pm$3.40)	0.17	-0.03	-36.26	-95.74	-96.01	-96.43	- 96.57

Table 5: Percentage deviation in accuracy from the base model on CIFAR-100

The results shown on CIFAR-100 consolidate indications provided in MNIST and CIFAR-10. Firstly, quantization is the most universally applicable method for compressing a network while retaining functional similarity and accuracy across various architectures. Secondly, pruning is also an effective strategy for maintaining functional similarity even when there is a significant degradation in accuracy, and different architectures have different sensitivity to L1 global pruning. Finally, this last experiment confirms that knowledge distillation cannot be considered a compression method as it in no way approximates the function of the teacher. The student is as functionally dissimilar to the base model as the independent models are, and sometimes even less. Subsequently, while some architectures can realise accuracy gains due to the teacher loss, it never brings the model close to an approximation of the teacher model.

5 Limitations and Conclusion

5.1 Limitations

Given the computational constraints of this work, it was not possible to train any of the models for longer than 25 epochs on each dataset. While 25 epochs were sufficient for both MNIST and CIFAR-10, it did not allow for a high test accuracy on CIFAR-100; consequently, to further validate the results, it would be worthwhile to use more training epochs to fully observe the trend of the phenomena witnessed during this exploration. Additionally, it would be advantageous to examine knowledge distillation into a smaller architecture and validate if this also behaves as an independently trained model, as witnessed in this paper.

5.2 Conclusions

The subject of this work was to answer the research questions:

1. Does the use of Knowledge Distillation, Quantization or L1 Pruning result in compressed models that are functionally similar to the uncompressed models?
2. Which compression methods, if any, are the most efficacious for function preservation?
3. How does compression and function preservation scale across multiple architectures and datasets?

The results presented in this paper answer each of these questions while simultaneously creating a new paradigm for how compression methods can be considered. Firstly, we show that quantization and L1 pruning can be defined as compression methods, as the resulting compressed model strongly resembles the original base model, notwithstanding fluctuations in the accuracy from the base. Not only do quantization and L1 pruning cluster around the base model they are much closer than independent models are to one another, further validating that the function of the base model is maintained during the application of these compression methods. The same results show that knowledge distillation produces students as functionally dissimilar to their teacher as independently trained models. The implication is that to consider knowledge distillation, a compression method is inaccurate. Instead, knowledge distillation should be perceived as a form of regularised training, which is somewhat alluded to but never elaborated on in the original distillation paper [11]. Given that no functional knowledge is passed from the teacher to the student, it could also be strongly argued that the name knowledge-distillation is misleading and, therefore, should be revisited and explored to understand what is truly happening in the distillation process and whether it is even possible to "transfer knowledge" between neural networks as this work suggests that it is not.

Secondly, the paper shows the importance of compression methods for functional preservation. In instances where a robust model is going to be compressed through various compression mechanisms, it would only be suitable to use quantization followed by pruning instead of a knowledge distillation method. Given that knowledge, distillation does not produce a similar student and that nothing is seemingly learnt from the teacher, it would not serve to preserve a robust model, and it would just be better to train a model with less capacity over conducting knowledge distillation. The relationship between compression and similarity shows that there is a strong alignment between the two notions and that more needs to be done to research how to preserve robust models that are accurate and tolerant to compression, leveraging more efficient machine learning systems while not forgoing accurate and sturdy functional representations.

Concerning compression methods at scale and across architectures, the results show that the efficacy of a compression method can be witnessed across datasets on specific architectures. The results show that there are interesting trends within different architectures to their receptiveness to pruning; the deeper ResNet-50 architecture appeared to be the most resilient to largescale pruning; this is possible because the depth of the network allows for the signal to pass through regardless of a reduction of parameters. However, more work would be needed to understand the relationship between deep networks and L1 pruning capacity. Additionally, the results provide a potentially more efficient pathway to pruning by using less complex proxy datasets on specific architectures to gain insights on how it will transfer to more complex datasets. As a result, this will allow less resource expenditure to find the appropriate compression for an architecture at scale.

Finally, the paper underscores the importance of evaluating compression methods from a functional perspective to verify that a model and its knowledge are being compressed. Through this perspective, common misinterpretations of methods such as knowledge distillation can be unveiled, and a more nuanced understanding grounded in functional and theoretical arguments can be made, all while ensuring compression techniques are applied in an efficient and informed manner.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

[2] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):1–18, 2017.

[3] Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. Scalable methods for 8-bit training of neural networks. *Advances in neural information processing systems*, 31, 2018.

[4] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Yevgen Chebotar and Austin Waters. Distilling knowledge from ensembles of neural networks for speech recognition. In *Interspeech*, pages 3439–3443, 2016.

[6] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, and . Padlewski, et al. Scaling vision transformers to 22 billion parameters. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7480–7512. PMLR, 23–29 Jul 2023.

[7] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[8] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[9] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[12] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[13] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[15] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

[16] Aashiq Muhamed, Iman Keivanloo, Sujan Perera, James Mracek, Yi Xu, Qingjun Cui, Santosh Rajagopalan, Belinda Zeng, and Trishul Chilimbi. Ctr-bert: Cost-effective knowledge distillation for billion-parameter teacher models. In *NeurIPS Efficient Natural Language and Speech Processing Workshop*, 2021.

[17] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[18] Junpeng Wang, Liang Gou, Wei Zhang, Hao Yang, and Han-Wei Shen. Deepvid: Deep visual interpretation and diagnosis for image classifiers via knowledge distillation. *IEEE Transactions on Visualization and Computer Graphics*, 25(6):2168–2180, 2019.

[19] Carole-Jean Wu, David Brooks, Kevin Chen, and . Chen, et al. Machine learning at facebook: Understanding inference at the edge. In *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 331–344, 2019.

[20] Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4388–4403, 2022.

[21] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.

[22] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

6 Appendix

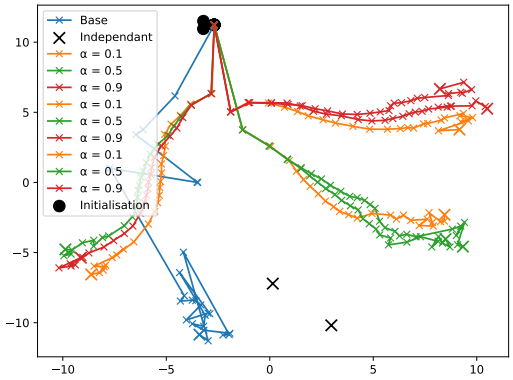


Figure 5: MNIST Functional Landscape for Self-Distillation on LeNet Architecture

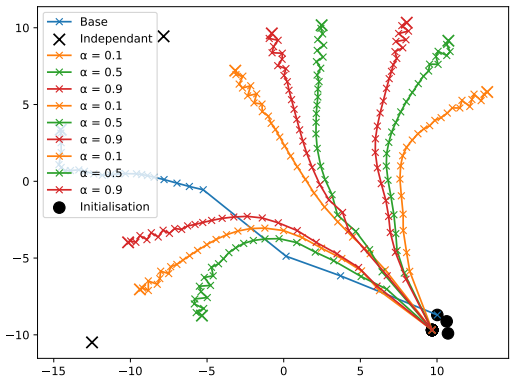


Figure 6: CIFAR-10 Functional Landscape for Self-Distillation on LeNet Architecture

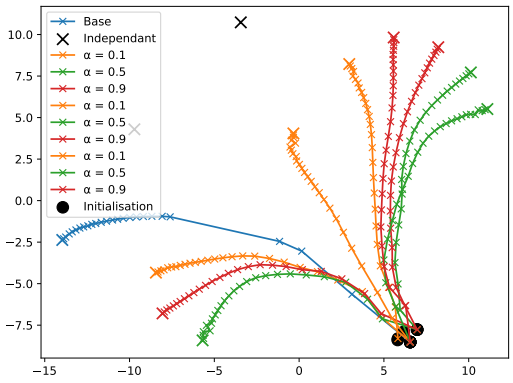


Figure 7: CIFAR-100 Functional Landscape for Self-Distillation on LeNet Architecture

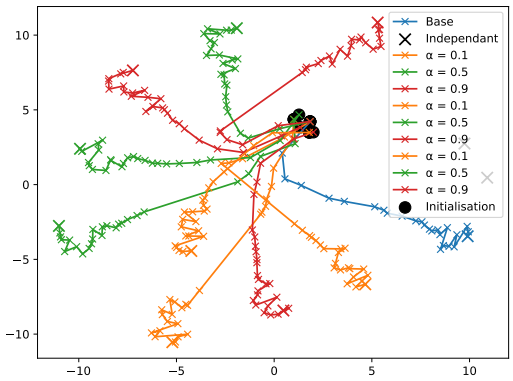


Figure 8: CIFAR-10 Functional Landscape for Self-Distillation on MobileNet Architecture

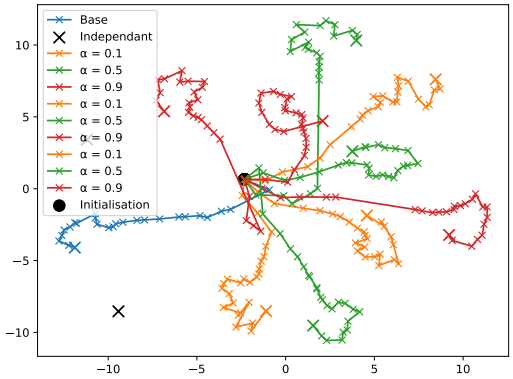


Figure 9: CIFAR-100 Functional Landscape for Self-Distillation on MobileNet Architecture

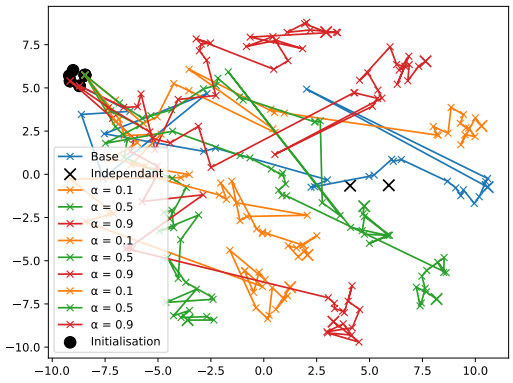


Figure 10: CIFAR-10 Functional Landscape for Self-Distillation on ResNet-50 Architecture