Empirical Study on How Deep Networks avoid the Curse of Dimensionality : Hierarchical Locality vs Weight Sharing

David Ha Eun Kang
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
haeun@mit.edu

Abstract

Both shallow and deep networks are universal, but it is the family of deep networks that avoids the *curse of dimensionality*. According to theoretical constructs by Poggio *et al.* [1], the exponential reduction of network complexity is permitted via *hierarchical locality*, not *weight sharing*. In this study, we run a set of experiments to test this theory, show further conditions for locality's triumph, and discuss why. On CIFAR-10, we do observe the dominant effect of kernel's hierarchical locality over weight sharing, and report that its exponential benefit is pronounced when SGD optimizer is used. The contrast is more evident when the total number of trainable parameters is set to near $500k$, compared to $50k$.

1 Motivation

1.1 Curse of dimensionality

Let $\sigma : R \to R$ be infinitely differentiable and not a polynomial. For $f \in W_m^n$, the complexity of shallow networks that guarantee accuracy at least ϵ is,

$$N = O(\epsilon^{-\frac{n}{m}}) \text{ and is the best possible}$$

Here, W_m^n is the set of all functions of n variables with continuous partial derivatives of orders up to $m < \infty$. The exponential dependence on the dimension n of the number $\epsilon^{-\frac{n}{m}}$ is known as the *curse of dimensionality*. Note that the degree of approximation is defined by $dist(f, V_N) = \inf_{P \in V_N} ||f - P||$. For example, if $dist(f, V_N) = O(N^{-\gamma})$ for some $\gamma > 0$, then a network with complexity $N = O(\epsilon^{\frac{1}{\gamma}})$ is sufficient to guarantee an approximation with accuracy at least ϵ.

1.2 Blessing of compositionality

Following the argument from Poggio *et al.* [1], deep networks avoid the curse of dimensionality, since it only requires network complexity of $O((n-1)\epsilon^{-\frac{2}{m}})$ to provide approximation with accuracy at least ϵ. Note that such exponential benefit is due to compositionality; functions composed of hierarchically local functions. That is, deep networks avoids the curse of dimensionality because of the blessing of compositionality via small effective dimension.

1.3 Role of weight sharing

One should note that weight sharing (a.k.a. invariance) also helps reduce the network complexity, but *not* exponentially as hierarchical locality does. In this study, we attempt to empirically show that while weight sharing helps, hierarchical locality is the main player.

2 Experimental setting

2.1 Architectures for locality vs. weight sharing

Experimental settings to compare the effect of locality and weight sharing is summarized in Table 1. For fair comparison, we matched the total number of parameters ($545,000 \pm 4\%$), stride (1), padding (valid, except for $k28$: details in section 3.3.), depth of hidden layers (4), and batch size (32).

Table 1: Experimental settings

Description	kernel size	hidden layers	stride	total params
Case 1. large kernel / weight sharing	28	4	1	545,108
Case 2. small kernel / no weight sharing	3	4	1	559,626
Intermediary. mid kernel / weight sharing	14	4	1	523,726
Benchmark. small kernel / weight sharing	3	4	1	553,420

2.2 Regularization

Data augmentation At each epoch, we generate new images from the original images by shifting 1/10 fraction of the total width and height, then apply random horizontal flip to add more bias to our model.

Figure 1: augmented images for enhanced learning

Batch normalization Batch normalization layer is attached after each convolutional (or conv-like) layers. The purpose is to maintain the mean output close to 0 and the standard deviation close to 1, which is known to be effective in smoothing the loss landscape and reduce the internal covariate shift.

Dropout Dropout layer is attached after each batch normalization layer. Dropout rate is set to be 0.1, which means elements in input is set to 0 with probability of 0.1. Remaining elements are scaled up by $1.0/(1.0 - rate)$ to preserve the expected value. Random assignment of 0 encourages each node to be independently useful, preventing overfitting.

2.3 Compute resources

A variety of AWS EC2 instances, including the M4(general purpose), C5 (compute optimized), R5 (memory optimized), and G4 (accelerated computing; NVIDIA T4 Tensor Core GPU), were configured using Amazon SageMaker. Also, NVIDIA's Tesla P100 data center GPU was used to accelerate the learning phase.

3 Results and Discussions

3.1 Main findings

"We report that the dominant effect of hierarchical locality of kernels over weight sharing is real, and that its exponential benefit is pronounced when SGD optimizer is used."

Figure 2. (a) illustrates our findings, where each legend corresponds to,

- Case 1 $\rightarrow$ WS_k28; weight sharing with large kernel of size 28
- Case 2 $\rightarrow$ NWS_k3; no weight sharing with small kernel of size 3
- Intermediary $\rightarrow$ WS_k14; weight sharing with medium kernel of size 14
- Benchmark $\rightarrow$ WS_k3; weight sharing with small kernel of size 3

As shown in Figure 2 (a), the blue arrow indicates the gain from locality (kernel size reduced from 28 down to 3), and the pink arrow indicates the gain from weight sharing. In both training and validation error on CIFAR-10 images, we observe that the locality gain wins!

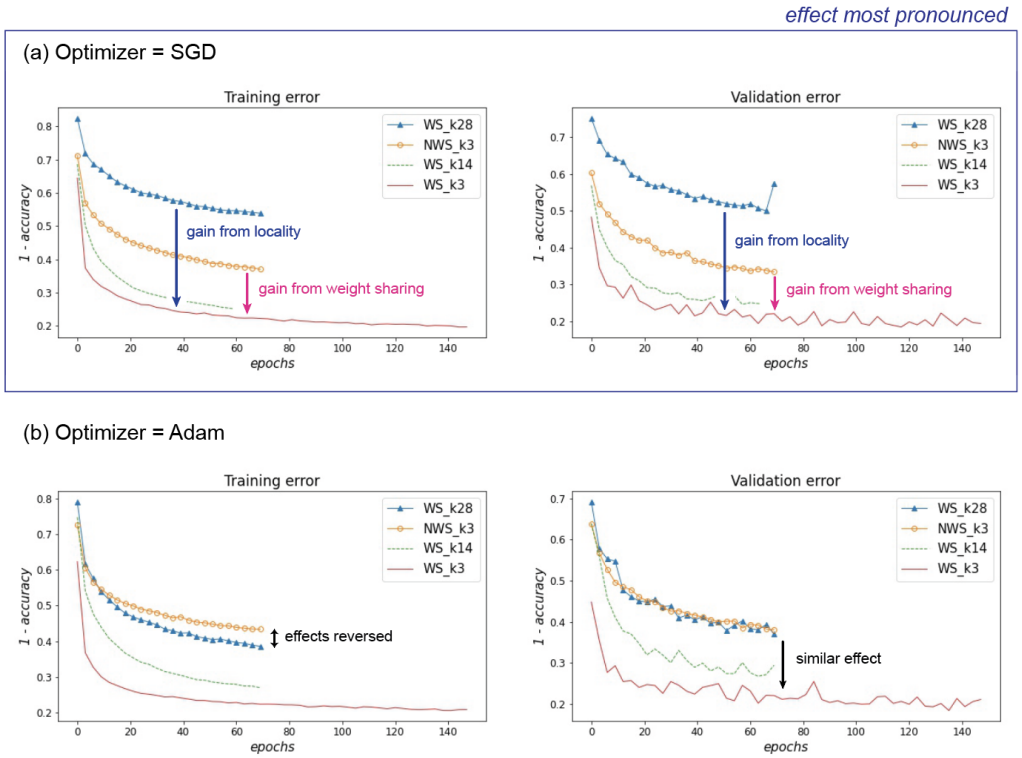

Figure 2: Locality vs weight sharing gain comparison for *Categorical Cross Entropy* loss on CIFAR-10. For fair comparison, we matched the total number of parameters ($545,000 \pm 4\%$), stride (1), padding (valid, except for $k28$: details in section 3.3.), depth of hidden layers (4), and batch size (32).

However, when the optimizer is set to Adam (Figure 2. (b)), the effects are reversed during training. When we switch the loss to Mean Squared Error and run experiments (Figure 3), we once again observe the larger locality gain for SGD, but big suppression happens again when Adam is used (Figure 3 (b)). It seems as if SGD is better at taking advantage of the hierarchical locality than Adam.

Why can't Adam take the same advantage of the locality gain? What may account for such misfortune can be found in the strength of the SGD algorithm. According to Hardt *et al.* [2], SGD is uniformly stable for even non-convex loss functions, and hence might have optimal generalization error, given that the number of iterations is not too large. On the other hand, the primary advantage of adaptive methods is the acceleration of training process that can converge faster, not the generalization itself

[3]. In this sense, the goal of SGD ("how to generalize well") and function approximation ("how to approximate functions well") are well aligned, hence we postulate that SGD enjoys the blessing of locality's exponential benefit ("providing easier approximation via locality") more than Adam.

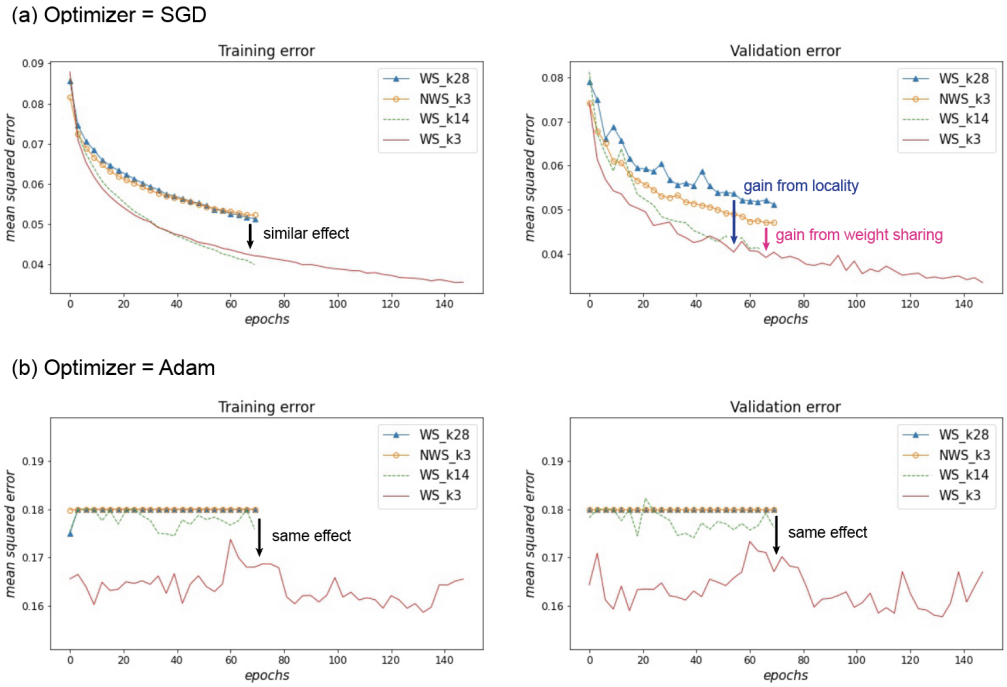

Figure 3: Locality vs weight sharing gain comparison for *Mean Squared Error* loss on CIFAR-10. For fair comparison, we matched the total number of parameters ($545,000 \pm 4\%$), stride (1), padding (valid, except for $k28$: details in section 3.3.), depth of hidden layers (4), and batch size (32).

3.2 Additional experiments

Next, we carry on another set of experiments with the best case (cross entropy loss with SGD), but this time, with reduced total number of trainable parameters ($56k$). This is an order of magnitude reduction from previous settings, where the new settings are summarized in Table 2 below.

Table 2: Experimental settings)

Description	kernel size	hidden layers	stride	total params
Case 1. large kernel / weight sharing	28	4	2	54,440
Case 2. small kernel / no weight sharing	2	4	2	58,542
Intermediary. mid kernel / weight sharing	11	4	2	57,262
Benchmark. small kernel / weight sharing	2	4	2	56,250

As shown in Figure 4, it is exciting to observe again both the gain from locality and from weight sharing! However, this time, the comparative advantage of locality is less pronounced at each epoch. We postulate that this is because with less parameters, it became more challenging to approximate the ground truth at the first place. Notice how low the accuracy is, compared to the case of $545k$ parameters, at each epoch.

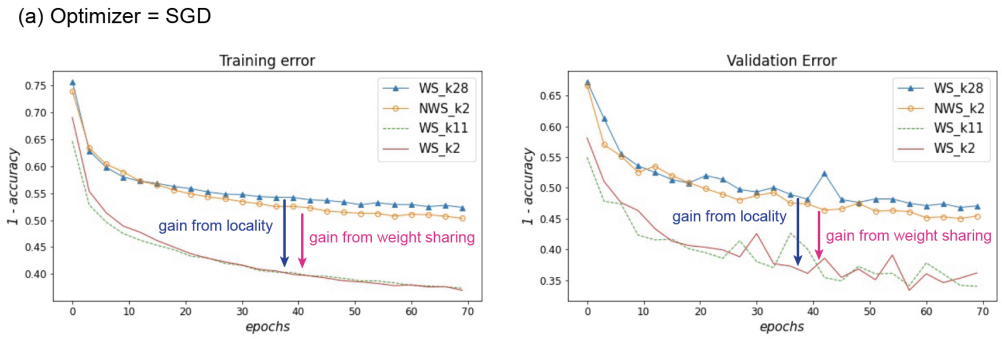

Figure 4: Locality vs weight sharing gain comparison for *Categorical Cross Entropy* loss on CIFAR-10. For fair comparison, we matched the total number of parameters ($56,000 \pm 5\%$), stride (2), padding (valid, except for $k28$: details in section 3.3.), depth of hidden layers (4), and batch size (32).

4 Conclusion

In short, from this empirical study, we conclude that,

1. *The dominant effect of hierarchical locality of kernels over weight sharing does exist.*
2. *Its exponential benefit is pronounced when SGD optimizer is used.*
3. *The effect is clear in the case of* $500k$ *number of parameters, more so than* $50k$*.*

5 Future works

Padding Note that we set the padding condition to 'same' for WS_k28 case, rather than 'valid'. There can be two ways of solving this mismatch. First is through updated library. Currently, the keras library only supports 'valid' condition for padding in LocallyConnected2D layers. When it starts to support 'same' condition at some later time, one can run the entire experiment with 'same' condition to control the effect of padding. Second way is through using padded images; one can manually code one padding layer and add it before each LocallyConnected2D layer.

Stripped images If we are able to create a ring or torus of images, i.e., images with edges stitched, we may be able to completely avoid the edge effects.

6 Acknowledgements

This project would not have been possible if without the great guidance and beautiful lectures of Professor Tomaso Poggio and helpful tips from Akshay Rangamani, MIT 9.520 senior staff.

References

[1] Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., Liao, B. (2017). Theory I: Why and When Can Deep Networks Avoid the Curse of Dimensionality? CMBB Memo No.058.

[2] Hardt, M., Recht, B., Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In International Conference on Machine Learning (pp. 1225–1234). PMLR.

[3] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. arXiv preprint arXiv:1705.08292.