A Theoretical-Empirical Approach to Estimating Sample Complexity of DNNs (Supplementary Material)

6. Derivations for Equation 2

$$
\begin{aligned}
\mathbb{E}_u^{\langle x_i, x_j\rangle}[\psi(\hat{x})] &= \int_{x_i}^{x_j} |\hat{x} - x(\hat{x})| u(\hat{x}) d\hat{x} = \int_{x_i}^{\frac{x_i+x_j}{2}} (\hat{x} - x_i) u(\hat{x}) d\hat{x} + \int_{\frac{x_i+x_j}{2}}^{x_j} (x_j - \hat{x}) u(\hat{x}) d\hat{x} \\
&= \int_{x_i}^{x_i+\frac{\rho(\hat{x})}{2}} (\hat{x} - x_i) u(\hat{x}) d\hat{x} + \int_{x_i+\frac{\rho(\hat{x})}{2}}^{x_i+\rho(\hat{x})} (x_i + \rho(\hat{x}) - \hat{x}) u(\hat{x}) d\hat{x} \\
&= \frac{1}{\rho(\hat{x})} \left(\int_{x_i}^{x_i+\frac{\rho(\hat{x})}{2}} (\hat{x} - x_i) d\hat{x} + \int_{x_i+\frac{\rho(\hat{x})}{2}}^{x_i+\rho(\hat{x})} (x_i + \rho(\hat{x}) - \hat{x}) d\hat{x} \right) \\
&= \frac{1}{\rho(\hat{x})} \left(\left[\frac{\hat{x}^2}{2} - x_i \hat{x} \right]_{x_i}^{x_i+\frac{\rho(\hat{x})}{2}} + \left[x_i \hat{x} + \rho(\hat{x}) \hat{x} - \frac{\hat{x}^2}{2} \right]_{x_i+\frac{\rho(\hat{x})}{2}}^{x_i+\rho(\hat{x})} \right) \\
&= \frac{1}{\rho(\hat{x})} \left(- \frac{x_i \rho(\hat{x})}{2} + \frac{x_i \rho(\hat{x})}{2} + \frac{\rho(\hat{x})^2}{2} + \frac{1}{2} \left(x_i + \frac{\rho(\hat{x})}{2} \right)^2 - \frac{x_i^2}{2} - \frac{(x_i + \rho(\hat{x}))^2}{2} \frac{1}{2} \left(x_i + \frac{\rho(\hat{x})}{2} \right)^2 \right) \\
&= \frac{1}{\rho(\hat{x})} \left(\frac{\rho(\hat{x})^2}{2} + \frac{x_i \rho(\hat{x})}{2} + \frac{\rho(\hat{x})^2}{8} - \frac{x_i^2}{2} - x_i \rho(\hat{x}) - \frac{\rho(\hat{x})^2}{2} + \frac{x_i^2}{2} + \frac{x_i \rho(\hat{x})}{2} + \frac{\rho(\hat{x})^2}{8} \right) \\
&= \frac{\rho(\hat{x})}{4}
\end{aligned}
$$

7. Real Data Experiments

7.1. Finding effective dimensionality

Data set	Architecture	Baseline	Bottleneck width (d')					
			1	2	3	4	5	6
MNIST	LeNet	0.992	0.407	0.971	0.986	0.988	0.989	0.989
CIFAR-10	ResNet18	0.955	0.842	0.945	0.950	0.954	0.948	0.953
	VGG16	0.930	0.886	0.926	0.926	0.928	0.925	0.927
CIFAR-100	ResNet18	0.791	0.172	0.590	0.699	0.729	0.739	0.735
	VGG16	0.712	0.161	0.610	0.672	0.683	0.689	0.695
ImageNet	ResNet 50	0.763	0.018	0.272	0.626	0.674	0.686	0.697
Udacity	CovNet	0.950	0.9786	0.983	0.992	0.991	0.989	0.992

Table 4. Test accuracy captured for baseline model (model without bottleneck trained on full training data set) and models with different widths ($d' = \{1, 2, 3, 4, 5, 6\}$) of the bottleneck.

We perform nearest neighbor classification on the low-dimensional features extracted from the bottleneck model described in Section 4.2.1. The effective dimensionality we find using nearest neighbor framework for different data sets is as follows: $d = 2$ for CIFAR-10 and MNIST, $d = 2/3$ for CIFAR-100 and $d = 3/4$ for ImageNet, and it is consistent with the effective dimensionality found in the bottleneck experiments.

Data set	Model	Baseline	Feature vector size					
			1	2	3	4	5	6
MNIST	LeNet	0.992	0.3922	0.970	0.986	0.987	0.988	0.989
CIFAR-10	ResNet18	0.955	0.838	0.944	0.950	0.952	0.951	0.952
	VGG16	0.930	0.798	0.923	0.924	0.921	0.923	0.927
CIFAR-100	ResNet18	0.791	0.205	0.579	0.664	0.688	0.709	0.718
	VGG16	0.712	0.169	0.575	0.643	0.645	0.668	0.677
ImageNet	ResNet-50	0.763	0.0153	0.311	0.585	0.626	0.642	0.649

Table 5. Nearest neighbor classification accuracy for different sizes of feature vectors ($d' = 1, 2, 3, 4, 5, 6$). Baseline model is the original DNN without bottleneck trained on full training data set.

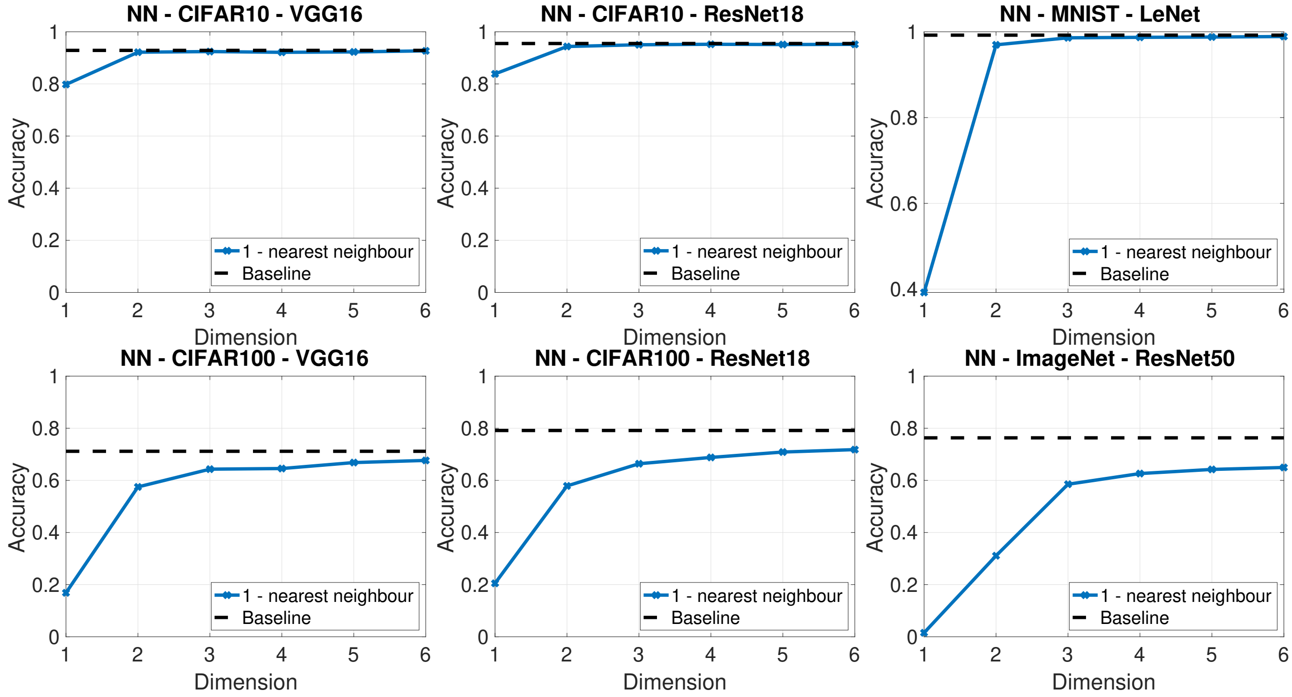

Figure 5. Nearest neighbor classification accuracy for different sizes of feature vectors ($d' = 1, 2, 3, 4, 5, 6$). Baseline model is the original DNN without bottleneck trained on full training data set and is marked with dashed line on the plots. **Top Row, from left to right:** VGG16 trained on CIFAR-10, ResNet-18 trained on CIFAR-10, and LeNet trained on MNIST, **Bottom Row, from left to right:** VGG16 trained on CIFAR-100, ResNet-18 trained on CIFAR-100, and ResNet-50 trained on ImageNet.

7.2. Learning Curves

Data	Model	Loss	BS	Opt	LR	LR decay (×0.1 at ep)	Weight decay
MNIST	LeNet	CE	128	SGD	0.01	-	0.0
CIFAR-10	ResNet18	CE	128	SGD	0.1	[150, 250]	0.0005
CIFAR-10	VGG16	CE	128	SGD	0.01	[150, 250]	0.0005
CIFAR-100	ResNet18	CE	128	SGD	0.1	[150, 250]	0.0005
CIFAR-100	VGG16	CE	128	SGD	0.01	[150, 250]	0.0005
ImageNet	ResNet-50	CE	256	SGD	0.1	[30,60,90]	0.0001
Udacity	CovNet	MSE	64	Adam	0.0002	[150]	0.0

Table 6. Hyper-parameters of different experiments. CE - Cross Entropy, MSE - Mean Square Error, Opt - Optimizer and LR - Learning Rate.

7.2.1 CIFAR-10

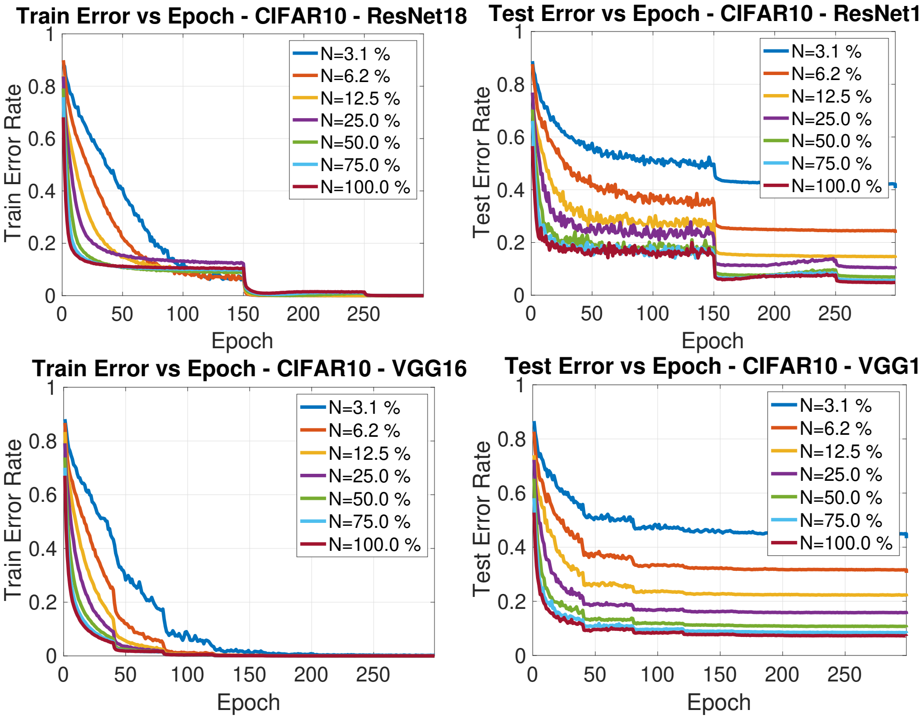

Figure 6. **(Left:)** Train and **(Right:)** test error vs number of epochs for **(Top:)** ResNet-18 **(Bottom:)** VGG16 models trained on increasingly larger subsets of CIFAR-10 data set.

7.2.2 CIFAR-100

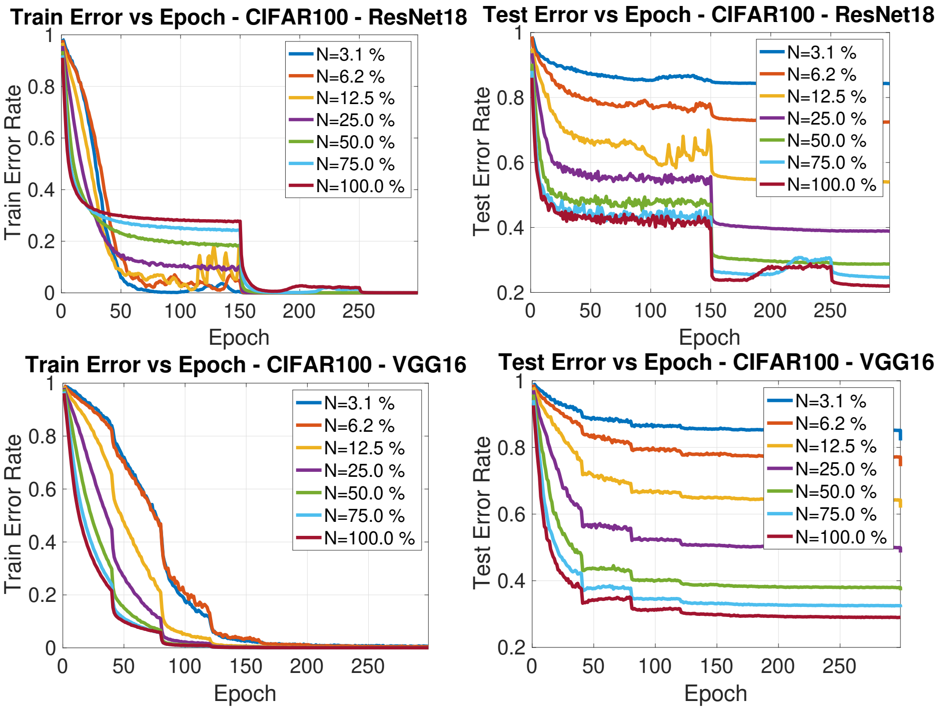

Figure 7. **(Left:)** Train and **(Right:)** test error vs number of epochs for **(Top:)** ResNet-18 **(Bottom:)** VGG16 models trained on increasingly larger subsets of CIFAR-100 data set.

7.2.3 ImageNet

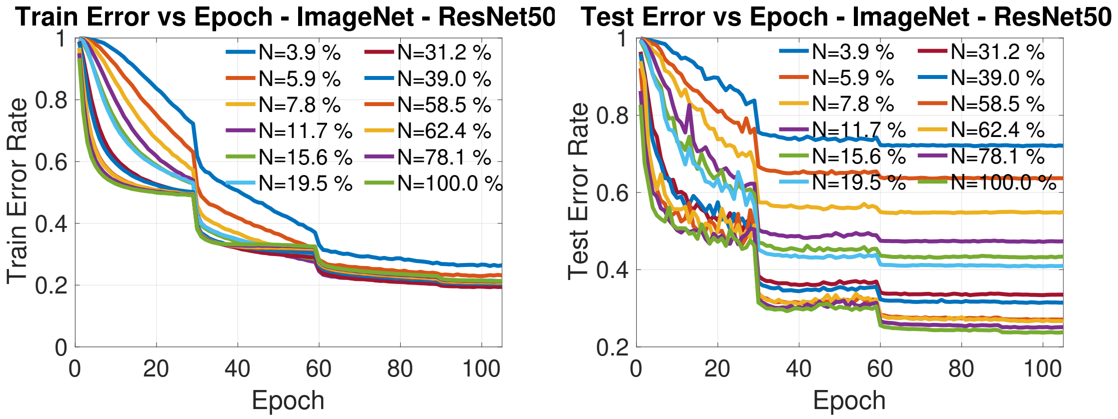

Figure 8. **(Left:)** Train and **(Right:)** test error vs number of epochs for ResNet-50 model trained on increasingly larger subsets of ImageNet data set.

7.2.4 MNIST

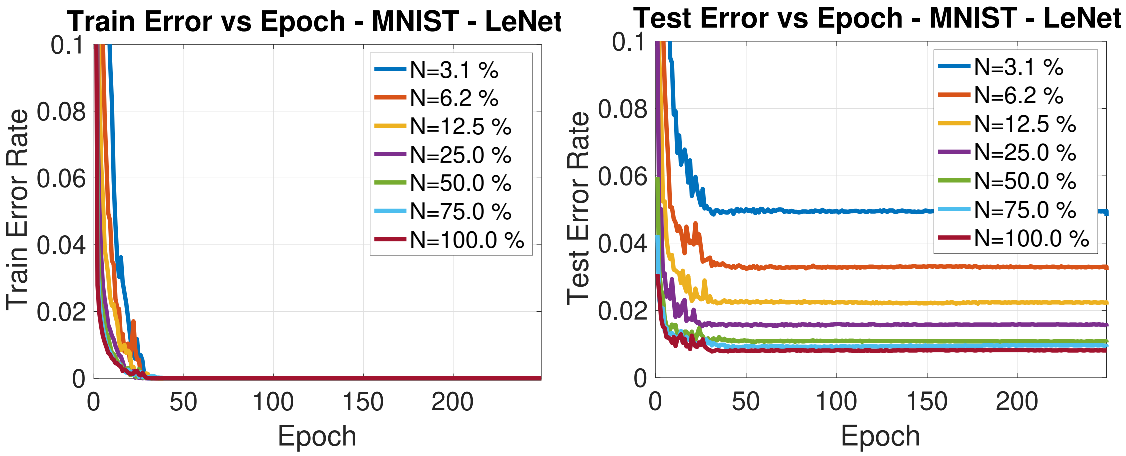

Figure 9. **(Left:)** Train and **(Right:)** test error vs number of epochs for LeNet models trained on increasingly larger subsets of MNIST data set.

7.2.5 Udacity

Layer	Output Size	Kernel	Stride	Padding
Conv	$32 \times 64 \times 64$	3×3	1	1
Conv	$64 \times 32 \times 32$	3×3	1	1
Conv	$128 \times 16 \times 16$	3×3	1	1
Conv	$128 \times 8 \times 8$	3×3	1	1
Linear	1024	-	-	-
Linear	1	-	-	-

Table 7. CNN architecture used in Udacity experiments. Each Convolution layer is followed by ReLU, Maxpool, and Dropout layers and each Linear layer is followed by ReLU and Dropout except the last linear layer.

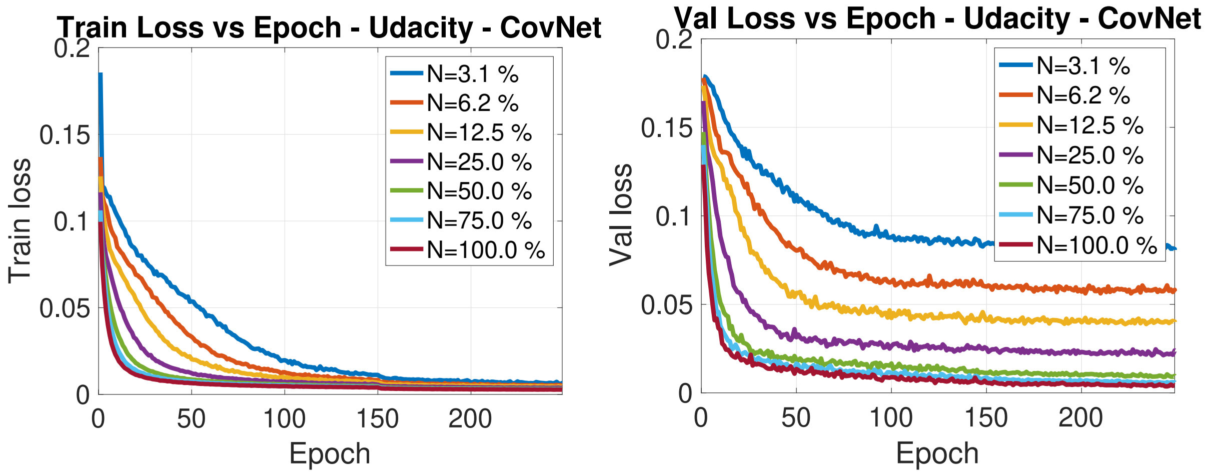

Figure 10. **(Left:)** Train and **(Right:)** test error vs number of epochs for CNN models trained on increasingly larger subsets of Udacity data set.

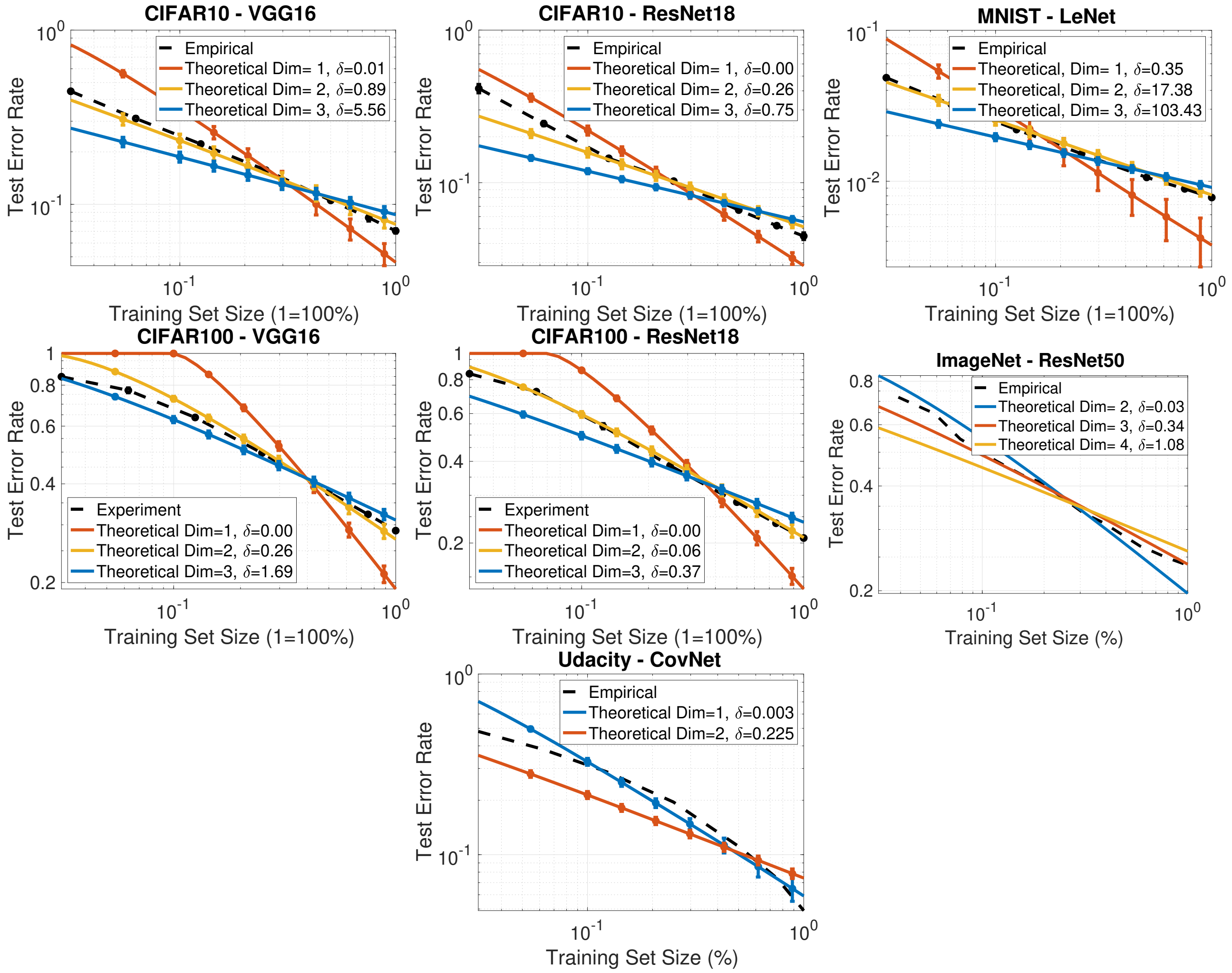

Figure 11. Empirical and theoretical learning curves (the latter obtained for different values of effective dimensionality).