Neural Dataset Generality - Supplementary

Ragav Venkatesan
Arizona State University
ragav.venkatesan@asu.edu

Vijetha Gattupalli
Arizona State University
jgattupa@asu.edu

Baoxin Li
Arizona State University
baoxin.li@asu.edu

1. Datasets Details

The setup we have used can be found in table 1. Although we chose only a handful of datasets, the intention of this article was only to show that such generality measures could be made. The scope of this article was not to benchmark various publicly available popular datasets. Neither was it to make suggestions specific to types of datasets.

2. Network architecture and learning details

We used one standard network architecture for all character datasets and experiments, one for Cifar 10 vs. Caltech 101 and another standard for Caltech 101 vs. Colonoscopy. The network architectures, learning rates and other details are provided in the sections below. The experiments were conducted on a Macbook Pro Laptop using an Nvidia GT 750M GPU, for character datasets and on an Nvidia Tesla K40 GPU for the others, with cuDNN v3 and Nvidia CUDA v7.

Table 1 shows the train-test-validation splits and the batch sizes used in stochastic gradient descent of all the datasets used. No preprocessing were done on the images themselves except for cropping, resizing, normalizing. The images were all normalized to lie in $[0, 1]$. The character recognition datasets were all of a constant $28X28$ grayscale, the Caltech 101 vs. Cifar 10 experiments were performed ar $32X32$, RGB and the Caltech 101 vs. Colonsoscopy were at $128X128$, RGB.

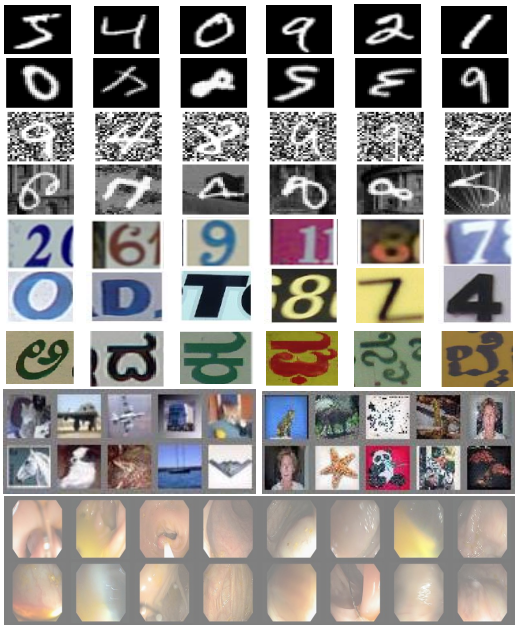

Figure 1. Samples of some of the datasets that we used in this analysis. From top to bottom: MNIST [1], MNIST-rotated [2], MNIST-random-background [2], MNIST-rotated-background [2], Google street view house numbers [3], Char 74k English [4], Char 74k Kannada [4]. Last two rows, first five from left are CIFAR 10 and the rest are Caltech101 [5, 6]. The bottom row is the colonoscopy dataset.

2.1. Architectures used for various experiments.

It is to be noted that the aim of the authors was not to set up the networks to achieve state-of-the-art. The authors did although try to achieve satisfactory performances on all base datasets involved before proceeding with the experimentation.

Character Datasets.

Our networks had three convolutional layers with 20, 20 and 50 kernels respectively. All the filters were 5 X 5 and were all stride 1 convolutions. The first layer didn't have any pooling. The second and the third layer maxpool by 2 subsampled. All the layers used rectified linear units ($ReLU$) activations [8]. The classifier layer was a softmax layer and we didn't use any fully connected layers. We used a dropout of 0.5 only from the last convolutional layer to the softmax layer [9]. We optimized a categorical cross-entropy loss using an rmsprop gradient descent algorithm [10]. For acceleration we used Polyak Momentum that linearly increases in range $[0.5, 1]$ from start to 100 epochs [11].

Dataset	Training	Testing	Validation	Classes	Training Batch Size
MNIST [1]	50,000	10,000	10,000	10	500
MNIST-random-background [2]	40,000	12,000	10,000	10	500
MNIST-rotated-background [2]	40,000	12,000	10,000	10	500
NIST Special Dataset-19 [7]	271,220	271,220	271,220	62	191
Google Street View House Numbers [3]	63,042	63,042	63,042	10	399
Char 74k English [4]	9,300	3,355	305	62	305
Char 74k Kannada [4]	5,694	1,314	1,752	100	438
MNIST $[4, 5, 8]$	14,000	2,500	2,500	3	500
MNIST $[0, 1, 2, 3, 6, 7, 9] - p$ per-class	$7p$	7,000	7,000	7	500
CIFAR 10 [5]	40,000	10,000	10,000	10	500
Caltech 101 [6]	5,080	3,048	1,016	102	254
Colonoscopy[1]	2,700	900	100	2	100

Table 1. Datasets used and their properties.

Unless early terminated, we ran 200 epochs. We also used a constant L_1 and L_2 regularizer co-efficients of 0.0001. Our learning rate was a 0.01 with a multiplicative decay of 0.0998.

CIFAR10 Vs. Caltech101 and Caltech 101 vs Colonoscopy.

For this task, the networks had five convolutional layers with $20, 20$, 50, 50 and 50 kernels respectively. We also had a last fully connected layer of 1800 nodes, which also had a dropout of 0.5. All the filters were 5 X 5 and were all stride 1 convolutions. Only the last layer maxpool by 2 subsampled. All the layers used rectified linear units ($ReLU$) activations [8]. All CNN and MLP layers were also batch normalized [12].The classifier layer was a softmax layer and we didn't use any fully connected layers. We used a dropout of 0.5 only from the last convolutional layer to the softmax layer [9]. We optimized a categorical cross-entropy loss using an rmsprop gradient descent algorithm [10]. For acceleration we used Polyak Momentum that linearly increases in range $[0.5, 0.85]$ from start to 100 epochs [11]. We use a learning rate of 0.001 for the first 150 epochs and then fine tune with a learning rate of 0.0001 for an additional 50 epochs unless early-terminated. Our learning decay of was subtractive 0.0005.

3. Additional Generality results.

Figure 2 shows generality curves that were not shown in the article.

References

[1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 1, 2

[2] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 473–480. 1, 2

[3] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*. Granada, Spain, 2011, vol. 2011, p. 5. 1, 2

[4] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, February 2009. 1, 2

[5] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009. 1, 2

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007. 1, 2

[7] P. J. Grother, "NIST Special Database 19 Handprinted Forms and Characters Database," 1995. 2

[8] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814. 1, 2

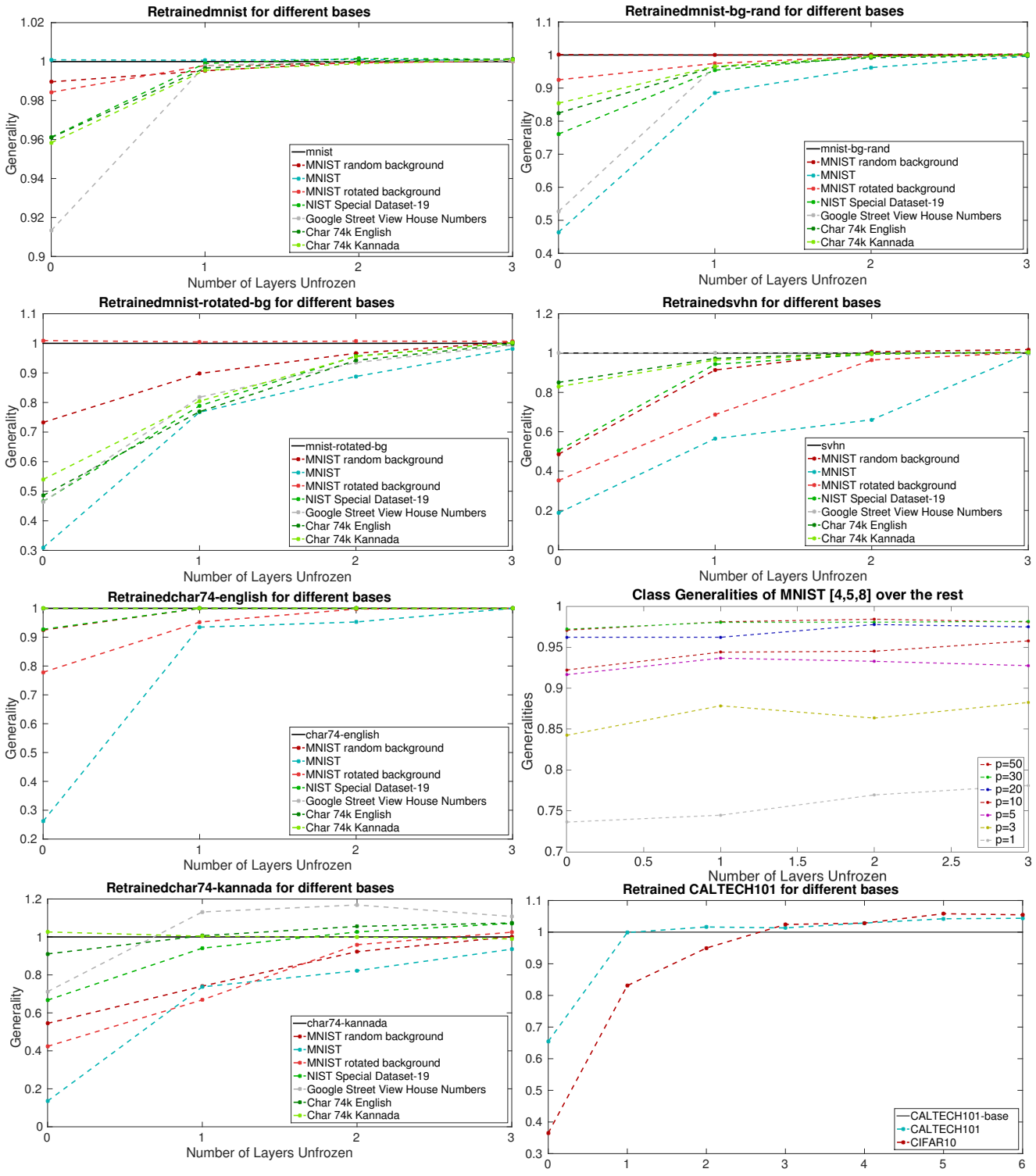

Figure 2. Generalities of datasets not shown in the actual paper. The dark line represents the accuracy of $n(D|r)$. Please zoom on a computer monitor for closer inspection.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014. 1, 2

[10] Yann N Dauphin, Harm de Vries, Junyoung Chung, and Yoshua Bengio, "Rmsprop and equilibrated adaptive learning rates for non-convex optimization," *arXiv preprint arXiv:1502.04390*, 2015. 1, 2

[11] Boris Teodorovich Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964. 1, 2

[12] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015. 2