

The Resistance to Label Noise in K -NN and DNN Depends on its Concentration - Supplementary Material

Amnon Drory
amnondrory@mail.tau.ac.il

Tel-Aviv University
Israel

Oria Ratzon
oriaratzon1@gmail.com

Shai Avidan
avidan@eng.tau.ac.il

Raja Giryes
raja@tauex.tau.ac.il

A Overview

This document contains supplementary material for the paper *The Resistance to Label Noise in K -NN and DNN Depends on its Concentration*. It contains the following:

- An experiment demonstrating the similarity between the softmax output of a network, and K -NN performed in the space of the outputs of its one-before-last layer.
- Derivation of algorithm for efficient calculation of the analytical expression for the expected accuracy of K -NN in the presence of label noise.
- Simplified analysis of the analytical expression for special cases.
- Additional implementation details.
- Diagrams demonstrating softmax. outputs for networks trained on various datasets with different noise models.

B Comparison of Softmax Outputs to K -NN Histograms

In this work, we have presented the conjecture that the output of the softmax layer tends to encapsulate the local distribution of the train examples in the vicinity of a given test example. In the main paper we demonstrate this by injecting noise into the training set, without having to explicitly define the space in which K -NN operates. Here, we demonstrate the similarity for a specific space: the 256-dimensional output of the penultimate layer of a network trained on clean data. We produce histograms of labels for K -Nearest Neighbors (with different values of K), and calculate the chi-square distance from these histograms to the softmax layer output.

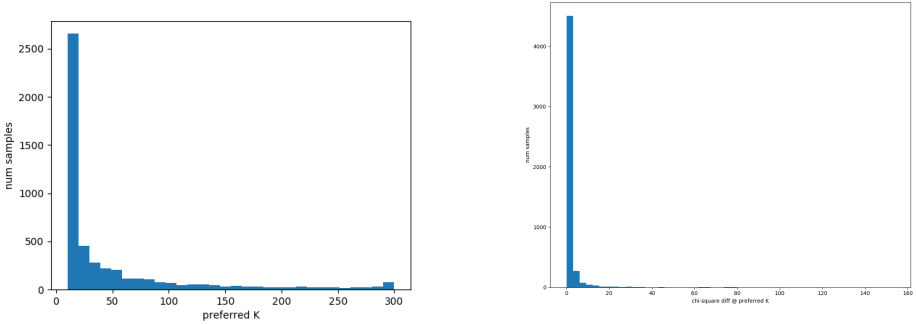


Figure 1: For each test example its *preferred K* is the value of K which yields the lowest chi-square distance to the softmax layer output. (a) shows a histogram of the prevalence of different choices of K . (b) shows a histogram of the chi-square distances, when each example is at its preferred K

The network used for the embedding space is trained on a clean version of the CIFAR10 dataset, and has the following structure: `cnv@20 - cnv@20 - pool - cnv@50 - cnv@50 - pool - fc@256 - fc@10 - softmax`,

where *cnv* is a convolutional layer using a 5x5 filter and zero-padding, *fc* is a fully connected layer, *@c* denotes the number of output channels, and *pool* is 2x2 max-pooling. Batch Normalization is added after each convolutional and fully-connected layer, followed by a ReLU non-linearity (except before the softmax layer). The features we use are the raw outputs of the fully connected layer with 256 output channels, before they are passed into batch normalization and ReLU. We try a range of K values, between 10 and 300, and for each example select its *preferred K value*, which is the one with the lowest chi-square distance. Fig. 1(a) shows the prevalence of different choices of K . Fig. 1(b) presents the histogram of the calculated chi-square distances.

The median chi-square distance between softmax layer output and K-NN histogram is 0.143123, which shows that the distributions are very close to each other. To get a better sense of the meaning of this number, we show a comparison of histograms for several examples in Fig. 2, where the chi-square distance is around this value. In each pair, the softmax output and the K-NN histogram for the example’s preferred K are presented. It can be seen that these histograms are very close to each other.

C Efficient Calculation of The Analytical Expression

We turn to present here an efficient strategy for computing the probability Q in Theorem 1 in the main paper. A naive computation of it, may iterate over all possible combinations of n_1, n_2, \dots , but only sum those where the plurality label is the correct one. As we shall see now, in addition to being inefficient, this is also unnecessary.

To make the calculation more efficient, we calculate the lower and upper boundaries of each n_i such that the summation only goes through the combinations that lead to a correct

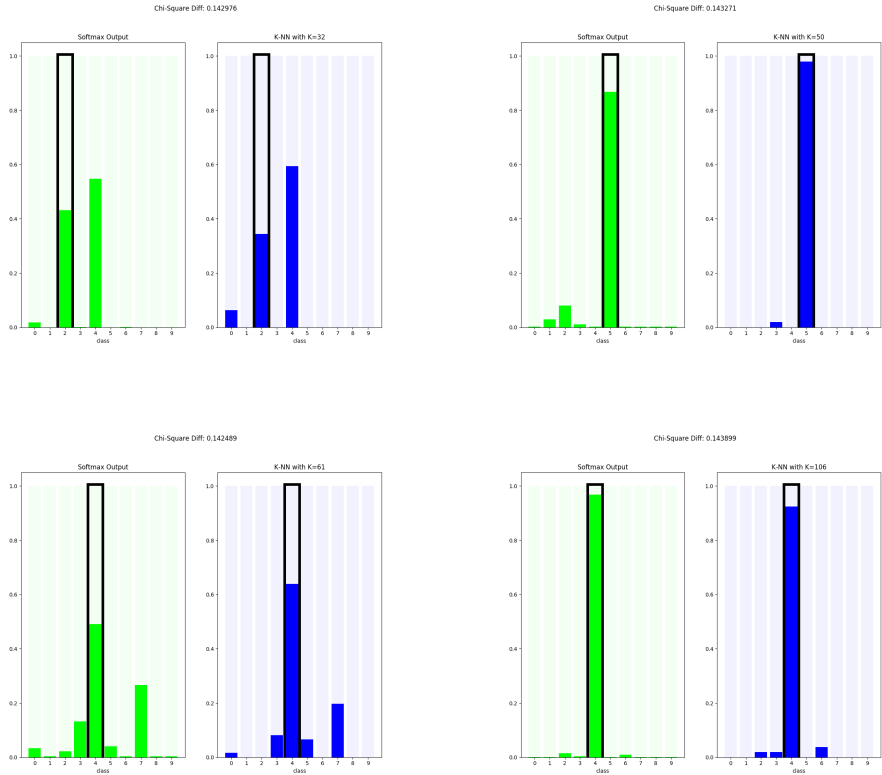


Figure 2: The median chi-square distance between softmax layer output and K-NN histogram is 0.143123. To get a sense of the meaning of this number, we show a comparison of histograms for several examples where the chi-square distance is around this value. In each pair the left (green) histogram is the softmax layer output, and the right (blue) is the K-NN histogram for the example’s preferred K.

plurality label. Denoting the lower bounds by m_i and the upper bounds by M_i , we have that

$$Q = \sum_{n_1=m_1}^{M_1} \sum_{n_2=m_2(n_1)}^{M_2(n_1)} \cdots \sum_{n_L=m_L(n_1, \dots, n_{L-1})}^{M_L(n_1, \dots, n_{L-1})} \binom{K}{n_1, n_2, \dots, n_L} q_1^{n_1} \cdot q_2^{n_2} \cdot \dots \cdot q_L^{n_L}, \quad (1)$$

where m_i is the smallest number of repeats of ℓ_i allowed, and M_i is the largest one. Their possible values are calculated in Section C.1. Notice that the number of repeats allowed for any label ℓ_i depends on the number of repeats already selected for all the previous labels, $\ell_j \forall j < i$.

For further efficiency, we can now decompose the summed expression so that shared parts of the calculation are only performed once. We decompose the multinomial coefficient into a product of binomial coefficients as follows:

$$\binom{K}{n_1, n_2, \dots, n_L} = \binom{K}{n_1} \cdot \binom{K-n_1}{n_2} \cdots \binom{K-\sum_{j=1}^{L-1} n_j}{n_L}, \quad (2)$$

and get the following formula for calculating Q :

$$Q = \sum_{n_1=m_1}^{M_1} \binom{K}{n_1} q_1^{n_1} \cdots \sum_{n_L=m_L}^{M_L} \binom{K-\sum_{j=1}^{L-1} n_j}{n_L} q_L^{n_L}. \quad (3)$$

C.1 Defining m_i and M_i

We will assume, without loss of generality, that the correct label is ℓ_1 . Clearly, we can repeat the same analysis by simply renaming or shuffling the labels. m_i and M_i need to be defined in a way that ensures:

1. There are exactly K letters in the string.
2. ℓ_1 is the plurality label, i.e. $n_1 > n_i \forall i \neq 1$.

We can start with M_1 , which is simply K . Clearly, a string consisting of K repeats of ℓ_1 fulfills both requirements. Once n_1 is known, we can define the maximum allowed number of repeats for any other letter as $M^* = n_1 - 1$. With the definition of M^* , we turn to calculate m_1 . Since $\sum_i n_i = K$ and $n_i \leq M^*$, we have that

$$K \leq n_1 + (L-1)M^* = n_1 + (L-1)(n_1 - 1). \quad (4)$$

By reordering the terms, we get that

$$n_1 \geq \frac{K + (L-1)}{L}. \quad (5)$$

Using the fact that m_1 is the smallest integer satisfying (5), we have

$$m_1 = \left\lceil \frac{K + (L-1)}{L} \right\rceil. \quad (6)$$

Having m_1 and M_1 set, we turn to calculate the values of $M_i \forall i \neq 1$. We start by defining R_i which is the number of string positions that are still unassigned:

$$R_i = K - \sum_{j=1}^{i-1} n_j. \quad (7)$$

Clearly, the value of n_i should be no larger than R_i . Thus,

$$M_i = \min\{R_i, M^*\}. \quad (8)$$

Lastly, we define m_i in a way that makes sure the string has no less than K letters:

$$m_i = \max\{0, R_i - (L - i) \cdot M^*\}. \quad (9)$$

The intuition here is that if all the subsequent letters $\ell_{i+1}, \dots, \ell_L$ have the maximal number of repeats, M^* , then ℓ_i need to be repeated enough times to bring the total repeats of all the yet unassigned letters to R_i .

D Simplified analysis of special cases

The process of calculating Q can be accelerated by several orders of magnitude if the following requirements are met:

1. The dataset is almost *perfectly learnable*, meaning that a CNN is able to reach approximately 100% test accuracy when trained with clean labels.
2. The conditional probabilities $P(\tilde{y}|y)$ are the same for all y , up to renaming of the labels.
3. The distribution of labels in the test set is *balanced*, meaning there is the same number of test examples for each label.

In these cases, the perfect learnability allows us to simplify C by assuming that for all train examples x , *all* clean labels in $\mathcal{N}(\hat{x})$ are the correct label:

$$C(\ell) = \begin{cases} 1 & \ell = \hat{y} \\ 0 & \text{else} \end{cases} \quad (10)$$

Also, the probability Q is the same for all test examples, from which follows $A_{K-NN} = Q$. For the *uniform noise* setting, q_j is simplified to

$$q_j = \begin{cases} (1-\gamma) + \frac{\gamma}{L} & \ell_j = \hat{y} \\ \frac{\gamma}{L} & \text{else,} \end{cases} \quad (11)$$

and for the *flip noise* setting, Q is simplified to

$$Q = \Pr(Y(\hat{x}) = \hat{y}) = \sum_{n=\lceil \frac{K+1}{2} \rceil}^K \binom{K}{n} \cdot (1-\gamma)^n \cdot \gamma^{K-n}, \quad (12)$$

where n is the number of examples in $\mathcal{N}(\hat{x})$ that have not been corrupted, and $K - n$ is the number of those that have been corrupted, i.e. flipped to the alternative label.

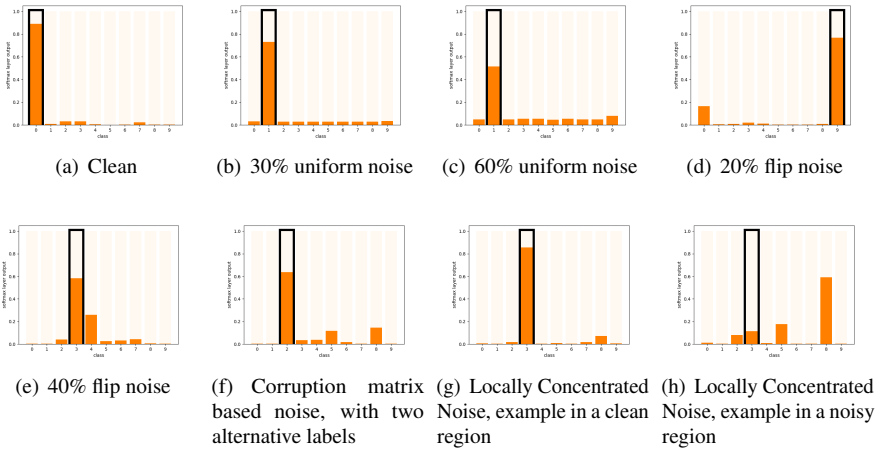


Figure 3: Softmax outputs of networks trained on noisy versions of the CIFAR-10 dataset. The ground truth label is marked by a black margin. Note that the network output tends to encapsulate the local distribution of labels in the vicinity of the input x .

E Additional Implementation Details

For all MNIST experiments, we use a DNN inspired by LENET-5 [1] and AlexNet [2], which reaches $\sim 100\%$ accuracy. Its structure is: $\text{cnn}@20 - \text{cnn}@20 - \text{pool} - \text{cnn}@50 - \text{cnn}@50 - \text{pool} - \text{fc}@FS - \text{fc}@10 - \text{softmax}$, where cnn is a convolutional layer using a 5×5 filter and zero-padding, fc is a fully connected layer, $@c$ denotes the number of output channels, and pool is 2×2 max-pooling. FS is 500 for Uniform Noise experiments and 256 for Flip Noise experiments. Batch Normalization [3] is added after each convolutional and fully-connected layer, followed by a ReLU non-linearity (except before the softmax layer).

Our data pre-processing in ImageNet training is inspired by ResNet [4]. Each image is resized so that its shorter side is changed to 256 (and the rest maintain the same aspect ratio). For training, we randomly sample a 224×224 crop from an image. For the test set we simply take the crop from the center of each image. As the network architecture, we use Densenet-121 [5] with Adam Optimization and mini-batch of size 256. The learning rate is initiated to 0.001 and then divided by 10 after 15 epochs. The models are trained up-to 30 epochs with early stopping.

For running-time considerations, The K -NN experiments on ImageNet were done using not the entire train set but instead a randomly selected subset of 2000 test examples (out of 50000 total).

References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [2] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional

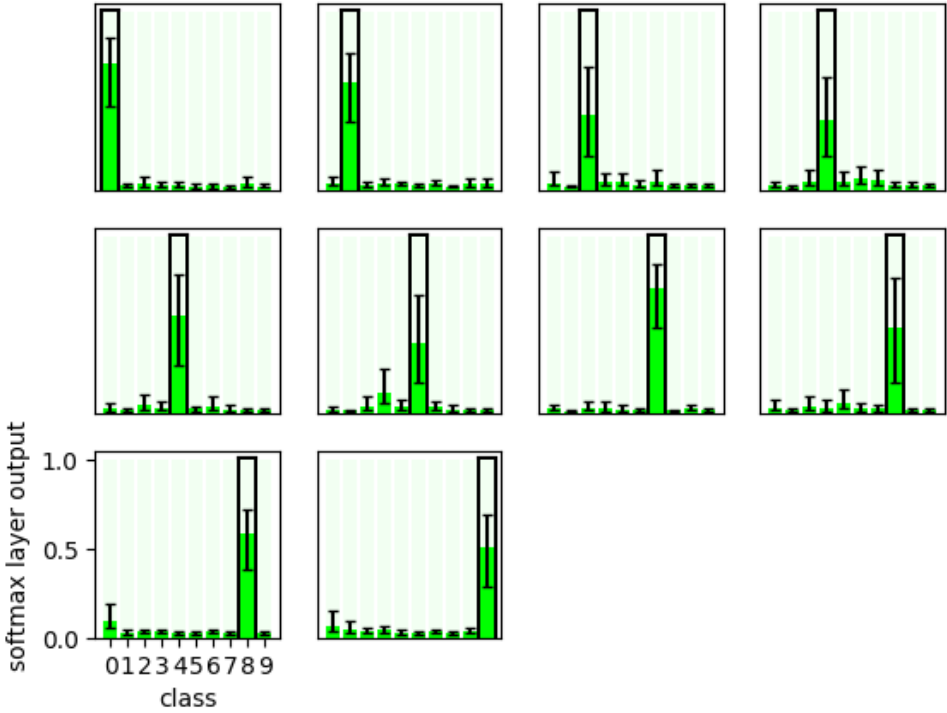


Figure 4: Detailed Aggregate Softmax outputs for a network trained on CIFAR-10 with 30% uniform noise. In each diagram, we show an aggregate of softmax vectors taken from all test examples that share the same ground truth label. In the top left diagram the GT label is 0, in the next diagram it is 1, etc. The height of the bars show the median, and the confidence interval shows the central 50% of examples. The ground truth label is marked by a black margin.

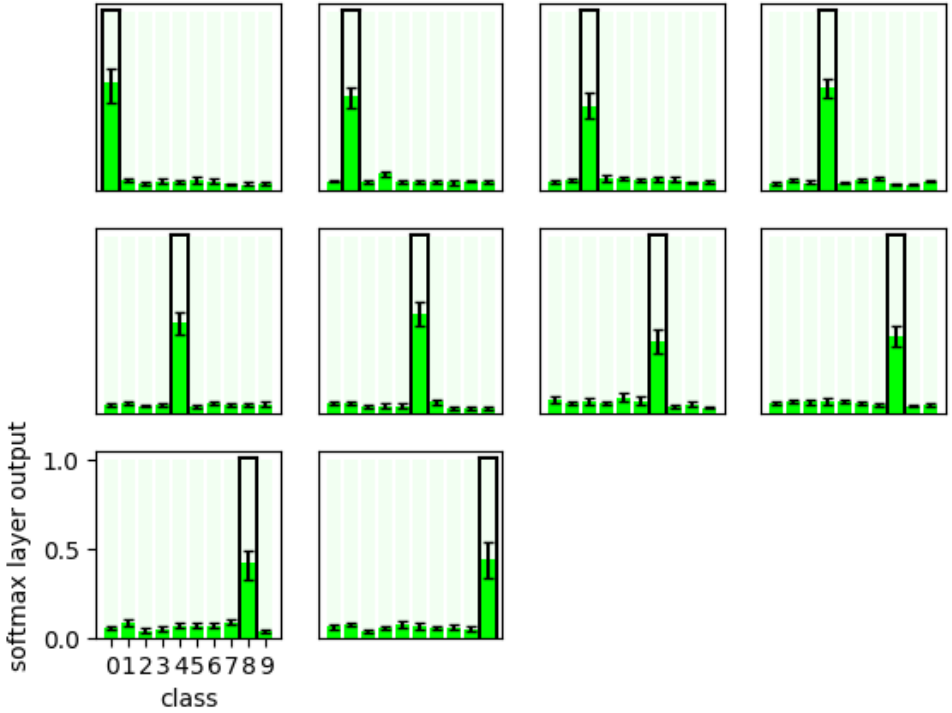


Figure 5: Detailed Aggregate Softmax outputs for a network trained on MNIST with 60% uniform noise. In each diagram, we show an aggregate of softmax vectors taken from all test examples that share the same ground truth label. In the top left diagram the GT label is 0, in the next diagram it is 1, etc. The height of the bars show the median, and the confidence interval shows the central 50% of examples. The ground truth label is marked by a black margin.

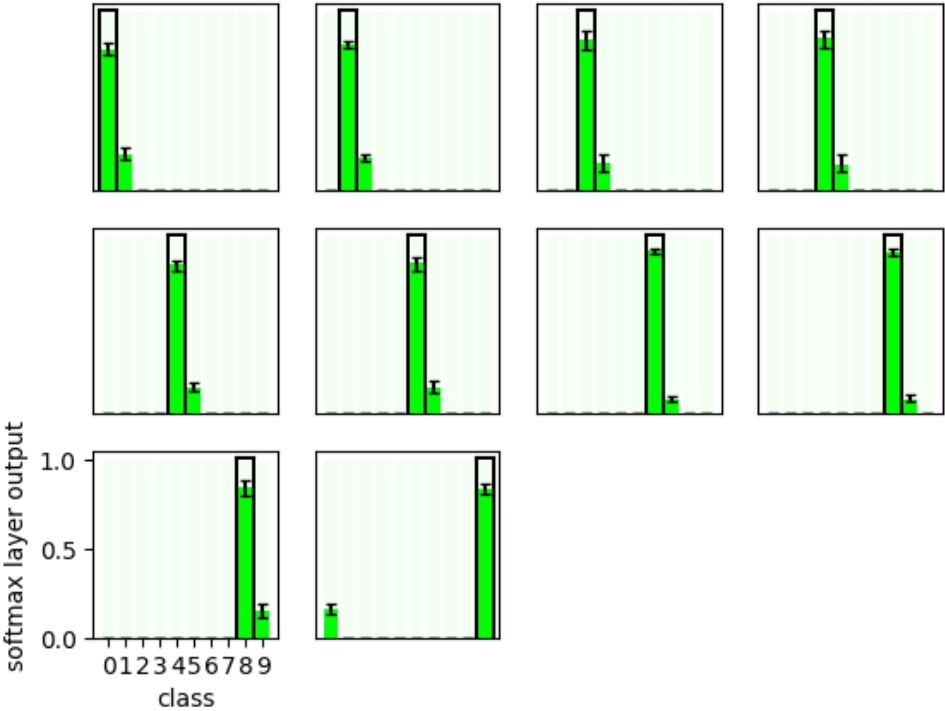


Figure 6: Detailed Aggregate Softmax outputs for a network trained on MNIST with 20% flip noise. In each diagram, we show an aggregate of softmax vectors taken from all test examples that share the same ground truth label. In the top left diagram the GT label is 0, in the next diagram it is 1, etc. The height of the bars show the median, and the confidence interval shows the central 50% of examples. The ground truth label is marked by a black margin.

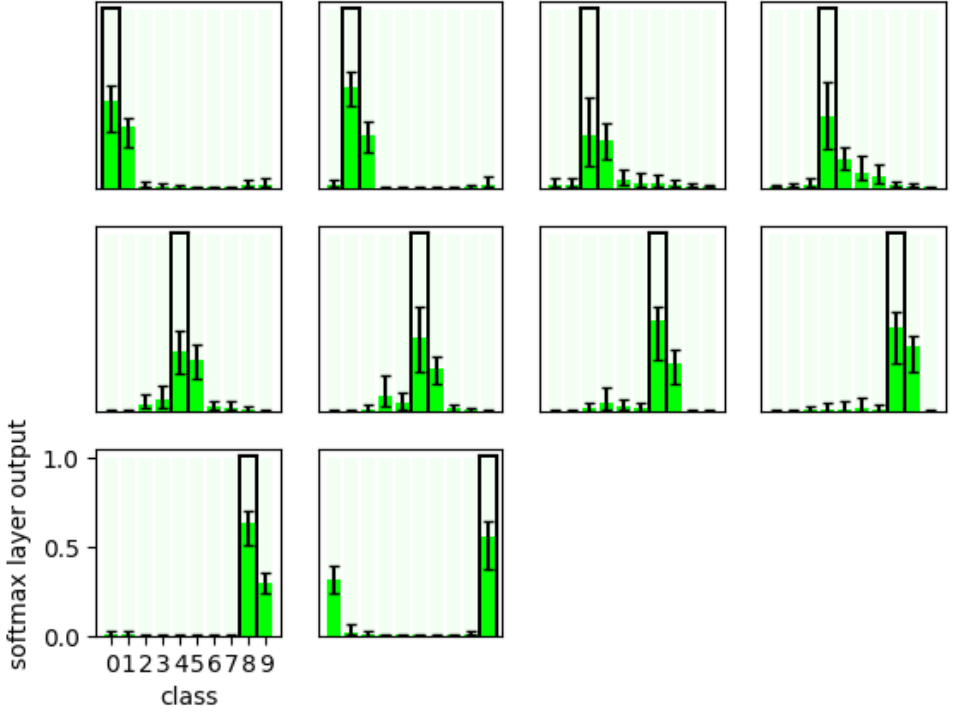


Figure 7: Detailed Aggregate Softmax outputs for a network trained on CIFAR with 40% flip noise. In each diagram, we show an aggregate of softmax vectors taken from all test examples that share the same ground truth label. In the top left diagram the GT label is 0, in the next diagram it is 1, etc. The height of the bars show the median, and the confidence interval shows the central 50% of examples. The ground truth label is marked by a black margin.

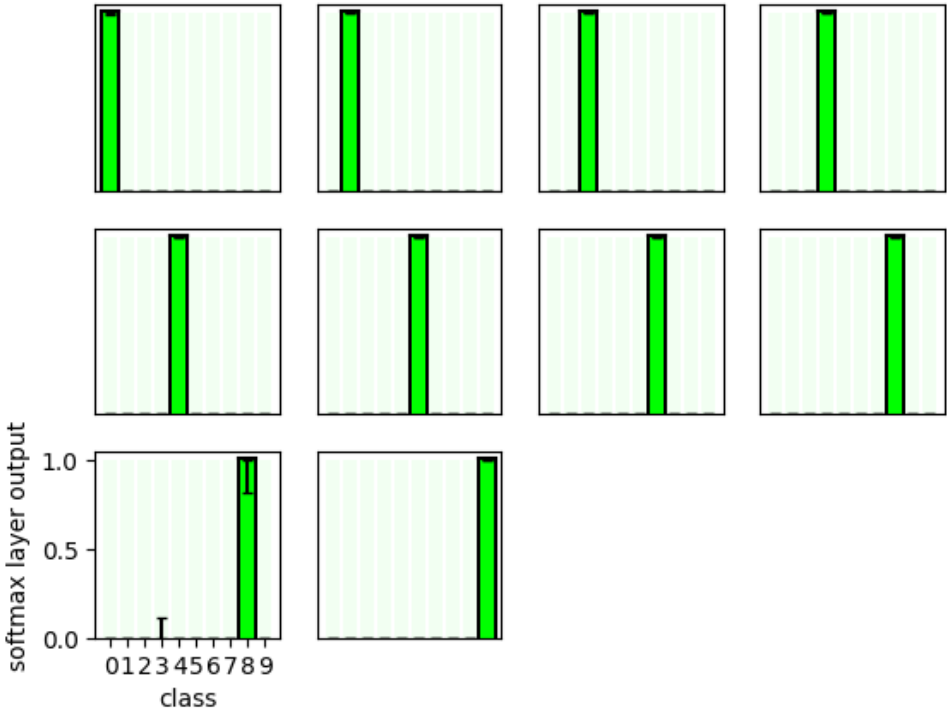


Figure 8: Detailed Aggregate Softmax outputs for a network trained on MNIST with 25% locally concentrated noise, with an example in the clean region. In each diagram, we show an aggregate of softmax vectors taken from all test examples that share the same ground truth label. In the top left diagram the GT label is 0, in the next diagram it is 1, etc. The height of the bars show the median, and the confidence interval shows the central 50% of examples. The ground truth label is marked by a black margin.

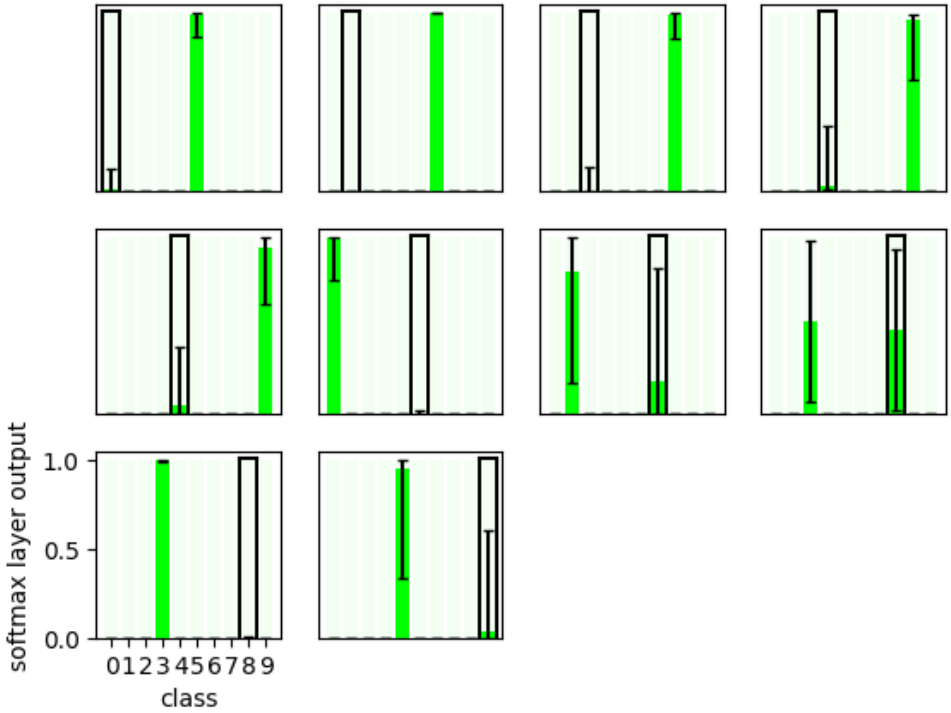


Figure 9: Detailed Aggregate Softmax outputs for a network trained on MNIST with 25% locally concentrated noise, with an example in the *noisy* region. In each diagram, we show an aggregate of softmax vectors taken from all test examples that share the same ground truth label. In the top left diagram the GT label is 0, in the next diagram it is 1, etc. The height of the bars show the median, and the confidence interval shows the central 50% of examples. The ground truth label is marked by a black margin.

- networks. *CoRR*, abs/1608.06993, 2016. URL <http://arxiv.org/abs/1608.06993>.
- [3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15*, pages 448–456. JMLR.org, 2015. URL <http://dl.acm.org/citation.cfm?id=3045118.3045167>.
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- [5] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.