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

Amnon Drory [1] **Oria Ratzon** [1] **Shai Avidan** [1] **Raja Giryes** [1]

Abstract

We investigate the classification performance of K-nearest neighbors (K-NN) and deep neural networks (DNNs) in the presence of label noise. We first show empirically that a DNN's prediction for a given test example depends on the labels of the training examples in its local neighborhood. This motivates us to derive a realizable analytic expression that approximates the multi-class K-NN classification error in the presence of label noise, which is of independent importance. We then suggest that the expression for K-NN may serve as a first-order approximation for the DNN error. Finally, we demonstrate empirically the proximity of the developed expression to the observed performance of DNN. Our result may explain an important factor in DNN robustness to label noise by showing that the less concentrated the noise the greater is the network resistance to it.

1. Introduction

Deep neural networks (DNN) provide state-of-the-art results in many applications. To train these models, large labeled data are required. Time and cost limitations come into play in their creation, which often results in imperfect labeling, or *label noise*, due to human error (Ipeirotis et al., 2010).

Perhaps surprisingly, it has been shown (Krause et al., 2015) that DNNs trained on datasets with high levels of label-noise may still attain accurate predictions. This phenomenon is not unique only to DNNs but also to classic classifiers such as K-Nearest Neighbours (K-NN) (Angluin & Laird, 1988; Natarajan et al., 2013; Frenay & Verleysen, 2014).

The starting point of this work is the observation that a DNN's prediction for a given test example depends on a local *neighborhood* of training examples. This is motivated by recent works that show that networks perform a smooth interpolation between the labels of training examples (Ongie et al., 2020; Savarese et al., 2019; Williams et al., 2019; Giryes, 2020). We suggest using K-NN as a first order approximation of such an interpolation. We validate this assumption and empirically demonstrate that the networks' last-layer output effectively encodes the distribution of training labels in such a neighborhood, i.e., provides a similar output to a K-NN applied on the embedding space of the network. This may suggest that bounds developed for a K-NN classifier applied to this space may apply to DNN. To this end, we develop an analytical expression that approximates the K-NN accuracy in the presence of randomly-spread label noise, which is of importance by itself, and show empirically that it matches the K-NN prediction accuracy.

Establishing the relationship between DNN and K-NN, we suggest that this bound may serve as a first order approximation to the accuracy of a network at different levels and types of noise. This relationship leads to an important conclusion about DNN resistance to label noise: The amount of resistance depends on how well the noisy examples are spread in the training set. When label noise is *randomly spread*, the resistance is high, since the probability of noisy examples overcoming the correct ones in any local neighborhood is small. However, when the noisy examples are *locally concentrated*, DNNs are unable to overcome the noise.

We validate our analytical expression for DNN using extensive experiments on several datasets: MNIST, CIFAR-10, and ImageNet. We show that empirical curves of accuracy-per-noise-level fit well with our mathematical expression. Proofs and more empirical results appear in the full paper.

2. Analysis of Robustness to Label Noise

The following steps are performed to analyze label noise: (i) we establish the different label noise models to consider; (ii) we show empirically that the output of the DNN's softmax resembles the label distribution of the K nearest training examples, linking DNNs to K-NN; and (iii) with this observation, we derive a formula for K-NN, which is of interest by itself, with the hypothesis that it applies also to DNN.

Setting. In the "ideal" classification setting, we have a training set $\mathcal{T} = \{x_i, y_i\}_{i=1}^{N}$ and a test set $\mathcal{S} = \{\hat{x}_i, \hat{y}_i\}_{i=1}^{M}$, where x is typically an image, and y is a label from the label set $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_L\}$. A classification algorithm (DNN or K-NN) learns from $\mathcal{T}$ and is tested on $\mathcal{S}$. The

[1] School of Electrical Engineering, Tel Aviv University. Correspondence to: Amnon Drory <amnon.drory@tau.ac.il>.

setting with label noise is similar, except that the classifier learns from a *noisy* training set $\{x_i, \tilde{y}_i\}_{i=1}^N$, which is derived from the clean data $\mathcal{T}$ by changing some of the labels. We designate by γ the fraction of training examples that are *corrupted* (with changed labels).[1]

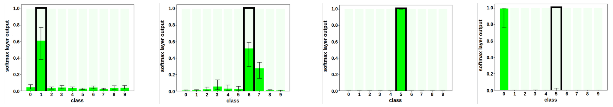
A common noise setting is that of *randomly-spread* noise. In this setting the process of selecting the noisy label $\tilde{y}$ is agnostic to the content of the image x, and instead only depends (stochastically) on the clean label y. The examples that get corrupted (i.e. their labels are changed) are selected uniformly at random from the training set $\mathcal{S}$. For each such example, the noisy label is stochastically selected according to a conditional probability $P(\tilde{y}|y)$ (we refer to this as the *corruption matrix*).[2] This setting can capture the overall similarity in appearance between categories of images, which leads to error in labeling.

Two simple variants of noise are often considered: Uniform Noise, and Flip-Noise. *Uniform Noise* is the case where the noisy label is selected uniformly at random from $\mathcal{L}$. This corresponds to a corruption matrix where $P(\tilde{y}|y) = \frac{1}{L}$ for all $\tilde{y}, y$. In the *flip label-noise* setting, each label ℓ_i has one counterpart ℓ_j with which it may be replaced. In this case the corruption matrix is a permutation matrix.

In contrast with the *randomly spread* setting, we also consider the *locally concentrated noise* setting, where the noisy labels are locally concentrated in the training set (Inouye et al., 2017). As an example, consider a task of labeling images as either *cat* or *dog*, and a human annotator that consistently marks all poodles as *cat*. We show that K-NN and, by extension DNN, are resilient to randomly spread label noise but not to locally concentrated one.

The connection between DNN and K-NN. We observe that DNN's prediction, similar to K-NN, tends to be the *plurality label* (most common) in a local neighborhood of train examples that surround the test example. The connection between K-NN and DNN is observed indirectly, by adding different types of noise to the training set, and analyzing its effect on the network's *softmax-layer output*. We find that this output tends to be the local probability distribution of the training examples in the vicinity of x: Its argmax $\ell_{pred} = \arg\max_{\ell \in \mathcal{L}} softmax_x(\ell)$ is the plurality label.

Fig. 1 presents the average softmax output of DNNs for various noise types and datasets. It demonstrates how the softmax layer output tends to be the distribution of the labels in the neighborhood of training examples. For example, when there is a uniform noise with noise level γ, we see that the peak value of the softmax is $1-\gamma + \frac{\gamma}{L}$ and the rest of the bins contain approximately $\frac{\gamma}{L}$, which is the number of noisy examples from each class expected to be in any local neighborhood. In the case of flip noise, it can be seen that the softmax probabilities spread mostly at the classes with which the flip occurs. and that the value is roughly proportional to amount of noise. It follows that the network makes a wrong prediction only when the "wrong" class achieves plurality in a local neighborhood. This, for example, is the case when locally concentrated noise is added and the test example is taken from the noisy region.



(a) CIFAR-10, 30% uniform noise (b) CIFAR-10, 40% flip noise (c) MNIST, Locally concentrated noise, example in a clean region (d) MNIST, Locally concentrated noise, example in a noisy region

Figure 1. **Softmax analysis:** Each diagram is aggregated from many test examples. The height of the bars shows the median, and the confidence interval shows the central 50% of examples. The ground truth label is marked by a black margin.

These findings provide us with an intuition into how DNNs are able to overcome label noise: Only the *plurality label* in a neighborhood determines the output of the network. Therefore, adding label noise in a way that does not change the plurality label should not affect the network's prediction. As long as the noise is *randomly spread* in the training set, the plurality label is likely to remain unchanged. The higher the noise level, the more likely it is that a *plurality label switch* will occur in some neighborhoods. When the noise type and noise level are known, it may be possible to produce a mathematical expression for the K-NN model, that predicts the probability of a switch. We suggest that this can serve as a first-order approximation to the behaviour of DNNs in the presence of noise. We show empirically that indeed it does match the observed behaviour of DNNs quite well in some settings. We also believe that this expression for K-NN is of independent interest, as it improves and extends previously known mathematical models for the resistance of KNNs to noise (Okamoto & Satoh, 1995).

K-NN accuracy in the presence of label noise. We turn to produce an analytical expression for the probability of a plurality switch, in the *randomly-spread* noise setting. Later, we also discuss the *locally concentrated* noise setting. We model randomly spread noise as follows: each test example $(\hat{x}_s, \hat{y}_s)$ has a local neighborhood $\mathcal{N}(\hat{x}_s)$ of K training examples. q_i is the probability for any example in $\mathcal{N}(\hat{x}_s)$ to have the *observed* label ℓ_i. The distribution q encodes the

[1] Note that the subset of "corrupted" examples may contain examples whose label has not changed if the randomly selected noisy label is the same as the original label.

[2] A *confusion matrix* C can be derived from the corruption matrix by $C = (1-\gamma)I + \gamma P$, where I is the identity.

results of the noise-creation process, and it depends on the parameters of this process, and on the clean labels of the examples in $\mathcal{N}(\hat{x}_s)$. Following (Okamoto & Satoh, 1995) we considerably simplify our derivation by introducing a small approximation: instead of treating the clean training examples as constant, we consider them to be sampled i.i.d from a *clean distribution* $C_s(\ell)$. The K-NN algorithm's prediction, which we denote by $Y(\hat{x}_s)$, is the plurality label. Its expected accuracy is defined as follows.

Definition 1 (K-NN Prediction Accuracy)**.**

$$A_{K-NN} \triangleq \frac{1}{M}\sum_{s=1}^{M} \Pr\big(Y(\hat{x}_s) = \hat{y}_s\big), \qquad (1)$$

where $\Pr\big(Y(\hat{x}_s) = \hat{y}_s\big)$ *is the probability that the plurality label of test example* $\hat{x}$ *in* $\mathcal{N}(\hat{x})$ *is the same as the ground truth label for* $\hat{x}$*.*

By expanding Eq. (1), we obtain an analytical formula for the accuracy of a K-NN classifier, which is given in the following theorem (proof based on combinatorial principles):

Theorem 1 (Plurality Accuracy)**.** *The probability of the plurality label being correct is*

$$Q \triangleq \Pr\big(Y(\hat{x}) = \hat{y}\big) = \sum_{n_1}\sum_{n_2}\cdots\sum_{n_L} [\![n_i > n_j,\ \forall j \neq i]\!] \cdot \binom{K}{n_1, n_2, \ldots, n_L} \cdot q_1^{n_1}\cdots q_L^{n_L}, \qquad (2)$$

where $[\![\cdot]\!]$ *is the indicator function,* $\hat{y} = \ell_i$ *is the correct label,* n_j *is the number of appearances of the label* ℓ_j *in* $\mathcal{N}(\hat{x})$ *and* q_j *is the probability of any such appearance.*

What is left to show is how to calculate q_j. The probability q_j is derived from the process that creates the noisy training set. Let $\hat{x}_s$ be a test example, and let x be a training example in $\mathcal{N}(\hat{x}_s)$. Let y be the clean label of x and $\tilde{y}$ be its noisy label. We denote by $C_s(\ell)$ the *clean label distribution* in $\mathcal{N}(\hat{x}_s)$. In other words, $C_s(\ell) \triangleq Pr(y = \ell)$. Thus, the expression for $q_j \triangleq \Pr(\tilde{y} = \ell_j)$ is

$$q_j = (1-\gamma)\cdot C_s(\ell_j) + \gamma \cdot \sum_{k=1}^{L} P(\ell_j|\ell_k)\cdot C_s(\ell_k), \qquad (3)$$

where γ is the noise level, and $P(\tilde{y}|y)$ is the corruption matrix that defines the corruption process. Eq. (3) shows that an example may be labeled with a label ℓ in two ways: Either it is uncorrupted and ℓ was its original label, or it was corrupted and received ℓ as its noisy label.

We can greatly improve the efficiency of calculating Q by first decomposing the multinomial coefficient into a product of binomials, and then decomposing Q into

$$Q = \sum_{n_1=m_1}^{M_1}\binom{K}{n_1} q_1^{n_1}\cdots\sum_{n_L=m_L}^{M_L}\binom{K-\sum\limits_{j=1}^{L-1} n_j}{n_L} q_L^{n_L}, \qquad (4)$$

where m_i is the smallest number of repeats of ℓ_i allowed, M_i is the largest, and together they encode the requirement that $n_i > n_j\ \forall j \neq i$. See supplementary material for a detailed derivation. Equation (4) contains many partial sums that are repeated multiple times, which allows further speedups by dynamic programming.

Estimating the clean distribution: In the K-NN setting, we can find the clean distribution by simply analyzing the clean data and noting the labels of the examples in the K-neighborhood of each test example. In the DNN setting, we can possibly do the same, using the one-before-last layer output as an embedding space in which to measure distances. Instead, we follow our observations in Fig. 1 and use the softmax layer output of a network trained on clean data. This results in a much more computationally efficient algorithm.

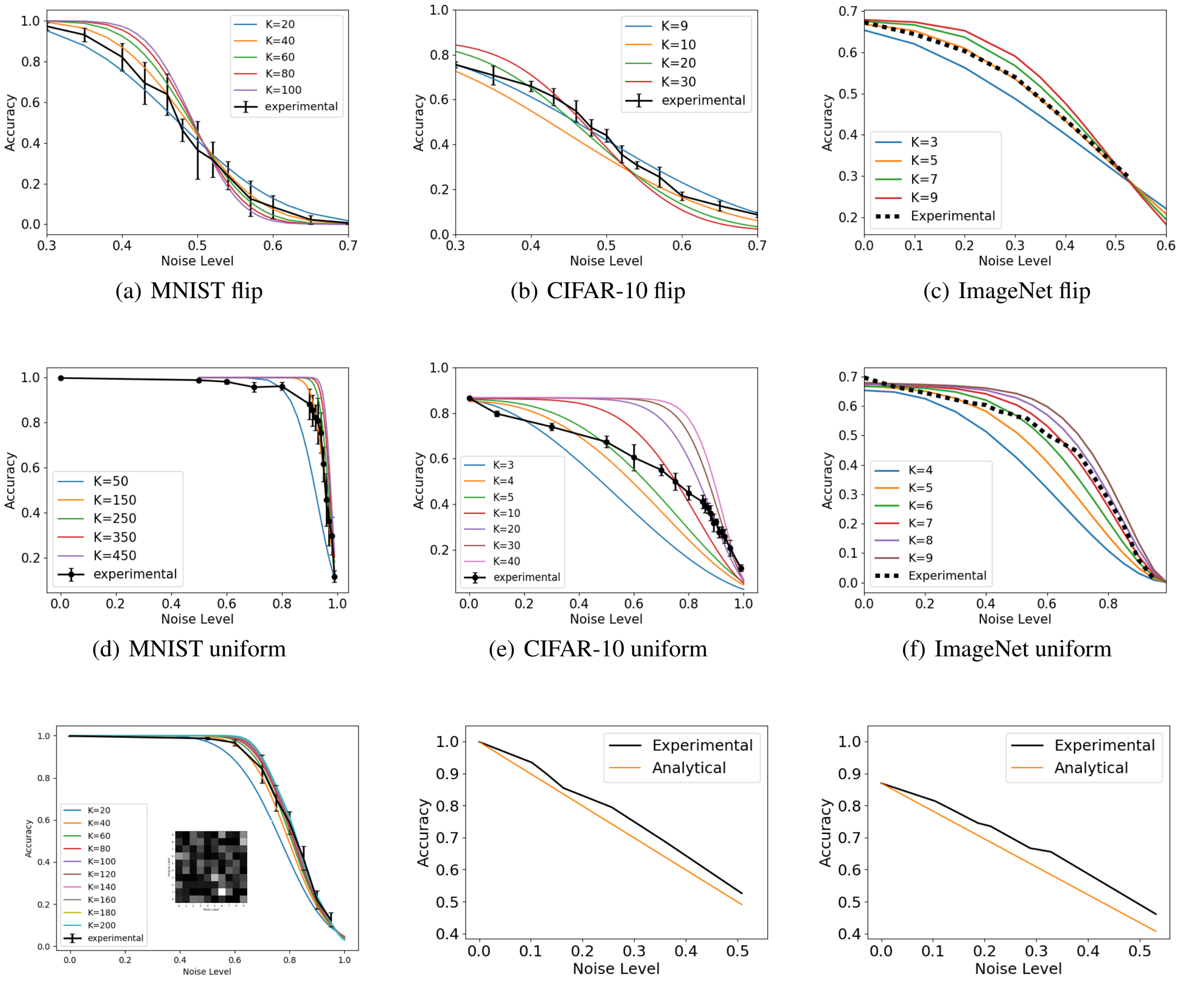
The locally-concentrated noise setting. An approximate analysis of DNN accuracy based on the K-NN algorithm can be done also in the locally concentrated noise setting. To do so, we need to assume that the noisy examples are concentrated in the feature space that K-NN operates in. If the noise is concentrated, then $\mathcal{N}(\hat{x})$ is almost always contained either in the *corrupt* area or *clean* area. In the first case, the prediction will be based on the corrupt label, therefore wrong. In the second, it will be correct. Therefore, the expected accuracy can be determined by the fraction of test examples for which $\mathcal{N}(\hat{x})$ is in the clean area. If we assume that the *test* examples are approximately uniformly spread in the example space, we can expect this fraction to be $1-\gamma$. Figs. 1(c,d) and Fig. 2(h,i) show this empirically.

3. Experiments

Our analytical model for K-NN acurracy in the presence of noise provides accuracy-vs-noise curves. We compare these to experimental curves derived from DNN trained on noisy data. We repeat these experiments with multiple noise types, and several popular datasets: MNIST, CIFAR-10, and ImageNet (ILSVRC 2012). Notice that we re-train the network for each dataset, noise type and noise level.

For all MNIST experiments, we use a DNN, which reaches $\sim$100% accuracy. For the CIFAR-10 experiments, we use the All Convolutional Network (Springenberg et al., 2014). To produce features for the K-NN experiments, an additional fully connected layer was added before the softmax, with 256 output channels. For ImageNet experiments, we use the Densenet-121 (Huang et al., 2016) architecture, with Adam Optimization and mini-batch of size 256. The feature used in K-NN experiments is 2048-dimensional.

The results of our experiments are summarized in Fig. 2. They contain four types of noise (uniform, flipped, general confusion matrix, and locally concentrated) and different values of K. We produce locally concentrated noise by



(a) MNIST flip
(b) CIFAR-10 flip
(c) ImageNet flip
(d) MNIST uniform
(e) CIFAR-10 uniform
(f) ImageNet uniform
(g) MNIST general corruption matrix
(h) MNIST concentrated noise
(i) CIFAR-10 concentrated noise

Figure 2. DNN Analytical and Experimental curves. The experimental curves show the mean accuracy and standard deviation. In most cases, the experimental curve is quite close to the corresponding analytical curves, and is clearly different from the analytical curves of the other settings (other subfigures). In (g) we also show the corruption-matrix $P(\tilde{y}|y)$ (rows are original label, columns are corrupt label, and brightness denotes probability, where white=high, black=low).

using k-means to find clusters of examples that are locally-concentrated in a feature space. In detail: we use the output of the penultimate layer of a network trained on clean data as a feature vector for each training example. In this space, we perform k-means for each class separately to divide it into k clusters. Then we select one of the clusters and change all of the labels in it into the same incorrect label.

The graphs in Fig. 2 show that we can calculate analytically the performance of the network for a given noise level, for some types of label noise. We plot the analytical curve (colored) for various K as we do not know the neighbourhood size. Yet, in all cases, the experimental curve (black) appears to naturally follow its corresponding family of analytical curves. We believe this indicates that the analytical curves approximate the *general behavior* of the experimental curves. In other words, our mathematical analysis captures a *major* factor in explaining the resistance of DNNs to spatially-spread noise. On a smaller scale, there are some deviations of the experimental curves from the anlytical ones. This could be caused by secondary factors that are not considered by the model.

The analytical expression predicts that DNNs may resist high levels of noise, but only if the noise is *randomly spread* in the training set (i.e., the uniform and flip settings). In contrast, in the locally concentrated noise setting DNNs are expected to have no resistance to noise. Note that indeed, this predicted behavior is demonstrated in the plots.

References

Angluin, D. and Laird, P. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Frenay, B. and Verleysen, M. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, May 2014.

Giryes, R. A function space analysis of finite neural networks with insights from sampling theory. *CoRR, abs/2004.06989*, 2020.

Huang, G., Liu, Z., and Weinberger, K. Q. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993.

Inouye, D., Ravikumar, P., Das, P., and Datta, A. Hyperparameter selection under localized label noise via corrupt validation. In *NIPS-LLD (Learning with Limited Data) Workshop*, 2017.

Ipeirotis, P. G., Provost, F., and Wang, J. Quality management on amazon me- chanical turk. In *ACM SIGKDD workshop on human computation*, pp. 64–67, 2010.

Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J., and Fei-Fei, L. The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition. *ArXiv e-prints*, November 2015.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari., A. Learning with noisy labels. In *NIPS*, pp. 1196–1204, 2013.

Okamoto, S. and Satoh, K. An average-case analysis of k-nearest neighbor classifier. In *Proceedings of the First International Conference on Case-Based Reasoning Research and Development*, ICCBR '95, pp. 253–264, Berlin, Heidelberg, 1995. Springer-Verlag. ISBN 3-540-60598-3. URL http://dl.acm.org/citation.cfm?id=646264.685911.

Ongie, G., Willett, R., Soudry, D., and Srebro, N. A function space view of bounded norm infinite width re{lu} nets: The multivariate case. In *International Conference on Learning Representations*, 2020.

Savarese, P., Evron, I., Soudry, D., and Srebro, N. How do infinite width bounded norm networks look in function space? In *Conference on Learning Theory*, pp. 2667–2690, 2019.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014. URL http://arxiv.org/abs/1412.6806.

Williams, F., Trager, M., Panozzo, D., Silva, C., Zorin, D., and Bruna, J. Gradient dynamics of shallow univariate relu networks. In *Advances in Neural Information Processing Systems 32*, pp. 8376–8385. 2019.