ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Temporal Ensembling-based Deep k-Nearest Neighbours for Learning with Noisy Labels

Alexandra-Ioana Albu *

Department of Computer Science, Babeş-Bolyai University
1 M. Kogalniceanu Street - Cluj-Napoca, Romania

Abstract.

Label noise can significantly affect the generalization of deep neural networks. Nevertheless, it is omnipresent in real world applications. This paper introduces an approach for identifying the samples from a dataset which are likely to have correct annotations. The proposed method computes the agreement of a sample with its nearest neighbours retrieved from the feature space provided by a neural network. We introduce a temporal ensembling strategy which takes into account the agreement scores obtained by a sample during previous training epochs. The superiority of our approach over several baselines is shown on image classification datasets.

1 Introduction

Training deep neural networks requires large datasets containing carefully annotated instances. While obtaining such high quality labels can be costly, annotations retrieved through automatic but imprecise methods are usually much cheaper. Due to this reason, robust learning in the presence of label noise has a great practical importance [1, 2].

One of the prominent research directions for learning with noisy labels is represented by sample selection methods, which aim to identify instances that are likely to have correct annotations in order to use only these samples during training [3, 4]. A popular selection technique is given by k-nearest neighbours (k-NN) filtering, which considers clean samples to be the instances that agree in terms of label with their k nearest neighbours [4, 5, 6]. This idea, which dates back to the work of Wilson [7], has been recently investigated in deep learning contexts, by retrieving neighbours using the features obtained from neural networks layers [5, 6]. In this paper, we propose a deep k-NN approach that takes into account neighbours obtained for more than just the current epoch in order to identify clean samples. The proposed approach employs a temporal ensembling [8] procedure for calculating the scores used to decide if a sample has a correct label. More specifically, a weighted average between past scores of a sample and the score obtained for the current epoch is computed. Our method aims to explore whether leveraging the information learned by deep

*This work was supported by a grant of the Romanian Ministry of Research, Innovation and Digitalization, CNCS/CCCDI-UEFISCDI, project number PN-III-P4-ID-PCE-2020-0800, within PNCDI III. The author would like to thank Dr. Luigi Malagò from Quaesta AI & Transylvanian Institute of Neuroscience (TINS) for his insightful feedback on the paper.

neural networks during training can improve k-NN filtering. We evaluate the proposed approach on benchmark image datasets.

2 Related work

The literature approaches for mitigating label noise include robust loss functions, regularizers, or sample selection strategies. The *generalized cross-entropy* (GCE) loss [9] combines the advantages of the non-robust cross-entropy (CE) and of the robust mean absolute error. Active-Passive losses (APL) [10] are robust losses which are formed using a normalization of a non-robust term and a robust term. Iscen et al. [1] introduced a regularizer that matches the prediction of a sample with the predictions of its neighbours. Several methods detect clean samples as the ones for which a small loss is obtained [3, 2], motivated by the observation that neural networks first learn the correct labels and only afterwards overfit the noisy ones. In [2], scores are assigned to small-loss samples using so-called local votes, that take into account only the current batch and global votes, which are computed using the training history. Co-teaching [3] is based on the small-loss selection strategy, but uses two networks which select clean instances for each other. k-NN methods include the work of Bahri et al. [4], which filters samples based on whether the sample's label agrees with the labels of its k nearest neighbours. The MOIT approach [5] includes a k-NN sample selection procedure that uses estimated labels for the samples neighbours and ensures that a balanced number of samples per class is selected. The approach proposed in [6] is another iterative method consisting of sample selection and relabelling using a balanced k-NN procedure.

3 Proposed Approach

Our proposed approach, Temporal Ensembling-based k-NN (TE-kNN) is a sample selection method built on a deep k-NN procedure [5, 4, 6]. TE-kNN is used iteratively during the training of a neural network on a dataset containing noisy labels in order to select a subset of reliable samples. These selected instances will be used to update the network's weights during the next epoch.

Given a sample x_i and its label $\mathbf{y}_i$ (represented as a one-hot vector) we compute the set $\mathcal{N}_i$ of its k nearest neighbours, considering the cosine distances between the representations obtained from the last layer in the neural network before the classification layer. The labels of these neighbours are used to estimate a probability distribution over the classes: $\mathbf{p}_i = \frac{1}{k} \sum_{j \in \mathcal{N}_i} \mathbf{y}_j$. The k-NN score of sample x_i for the current epoch is computed as the cross-entropy between the true label distribution and the label distribution estimated using the neighbours' annotations: $score_{kNN}(x_i) = -\mathbf{y}_i^T \log \mathbf{p}_i$. As noted in [5], lower scores indicate that the sample is more likely to have a correct label, since it agrees with the labels of its neighbours. However, the identified neighbours or their annotations may be incorrect due to the noisy labels used in training. Moreover, towards the end of training, the network is more prone to overfitting to incorrect examples than in early epochs [11, 3]. With the aim of obtaining more robust scores and inspired by the temporal ensembling strategy used in [8, 11], we propose to

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

compute a weighted average of the scores obtained for the previous epoch and the current k-NN scores. Unlike other k-NN sample selection approaches [4, 5, 6], we compute scores by taking into account the model's evolution during training. Therefore, the score for epoch t is obtained as: $score_t(x_i) = \alpha * score_{t-1}(x_i) + (1 - \alpha) * score_{kNN}(x_i)$. This strategy aims to improve the scores computations by considering both past and current scores. Our approach is different from the method introduced in [11], which applies temporal ensembling to obtain network predictions used for computing a new robust regularizer [11]. In contrast, we introduce a temporal ensembling procedure for estimating scores that measure how likely the label of a sample is to be correct.

Afterwards, $r(t)$% samples with the lowest scores are selected, where $r(t)$ can be obtained using various methods. In this study, we follow the strategy proposed by Han et al. [3]. Initially all available samples are used. $r(t)$ is linearly decreased during the first 10 epochs, up to the (known or estimated) proportion of correct labels present in the dataset. Then, a constant number of samples is selected. We note, however, that other strategies are possible, such as fitting a Gaussian mixture model on the obtained scores, as in [2] or defining adaptive per-class thresholds as in [5]. We study two variations of the proposed approach, by considering two types of initialization: zero-initialization and autoencoder (AE) initialization. The zero-initialization means that the samples scores are 0 at the beginning and the network is initialized with random weights. In the AE initialization, the network is initialized with the weights of the encoder component of an AE trained in an unsupervised manner on that dataset. In this setting, the initial samples scores are obtained by computing the nearest neighbours set in the latent space of the AE. The method is shown in Algorithm 1.

Algorithm 1: Training algorithm using the proposed TE-kNN method.

```
Input: dataset $\mathcal{D}$, initial sample scores $s_0$, hyper-parameter $\alpha$, number
       of epochs $n$, number of neighbours $k$, rates $r(t)$ for $1 \leq t \leq n$
Output: the trained model
$\mathcal{D}_{clean} \leftarrow \mathcal{D}$
$score_0 \leftarrow s_0$
for $t \leftarrow 1,n$ do
    @update model weights using the samples from $\mathcal{D}_{clean}$
    for $(x_i, y_i) \in \mathcal{D}$ do
        @compute nearest neighbours set $\mathcal{N}_i$ in the current feature space
         and $score_{kNN}(x_i)$
        $score_t(x_i) \leftarrow \alpha * score_{t-1}(x_i) + (1 - \alpha) * score_{kNN}(x_i)$
    end
    $\mathcal{D}_{clean} \leftarrow$ r(t)% samples with the smallest scores
end
```

4 Experimental Setup

Datasets Our approach was evaluated on the SVHN [12], CIFAR-10 and CIFAR-100 [13] datasets using synthetic label noise. The datasets are formed of RGB images grouped into 10 classes for SVHN and CIFAR-10 and into 100 classes in the case of CIFAR-100. The noisy training datasets are obtained following the procedure from [10], using so-called *symmetric noise*, by randomly swapping a part of the labels with a label that is incorrect. We generate two noisy versions of the dataset, using two noise rates previously investigated in the literature [5, 10]: 20% and 60%. The test sets contain the clean, original labels.

Training details We used a neural network architecture introduced in [10], which consists of six convolutional layers (filter sizes: 64-64-128-128-196-196) and two linear layers (neurons: 256–# of classes). Every layer in the network was followed by batch normalization and ReLU and after each group of two convolutions a max pooling layer was added. The batch size was set to 64 and the SGD optimizer with momentum was used, having an initial learning rate of 0.01. A cosine annealing procedure was applied to the learning rate, which was reduced up to a minimum value of 0.001. Weight decay of 10^{-4} was used. The network was trained for 120 epochs in the case of SVHN and CIFAR-10 and for 150 epochs on CIFAR-100. The hyper-parameters for TE-kNN were chosen using a clean validation set. k was set to 25 and α to 0.2 for CIFAR-10 and SVHN, while $k = 250$, $\alpha = 0.5$ for CIFAR-100. Weak data augmentations were applied, following [10]: random crops and horizontal flips for CIFAR-10 and SVHN and random crops, flips and rotations for CIFAR-100. The AE was formed of an encoder having the same architecture as the classifier (excluding the last layer) and a symmetric decoder. The AE was trained for 100 epochs using the Mean squared error loss and SGD with momentum and learning rate of 0.05. A learning rate scheduler and weight decay were applied for this model as well. The implementation was done using PyTorch.

5 Results and discussion

We compared our approach with multiple baselines: three loss functions - CE, APL [10], GCE [9] (see Section 2) - and two sample selection strategies - selection of the small loss instances and k-NN filtering (using only the current epoch's scores). For APL, we trained the NCE+RCE version, which gave the best overall results in [10]. The baselines were re-implemented using publicly available code [10, 5, 3, 6] and for GCE and APL we used the hyper-parameters suggested in [10]. All methods were trained in the same regime, using the same network and all sample selection methods used the same procedure for calculating the ratio of samples to be used in each epoch ($r(t)$), as described in Section 3. Table 1 shows the obtained results. We trained TE-kNN using the CE loss and zero-initialization (denoted in the table by TE-kNN) as well as using AE initialization (AE init). For a fair comparison with TE-kNN using AE init, we also report results for two baselines initialized with the pre-trained AE weights: the best competing loss function (APL) and the best competing sample selection method (k-NN). In both types of initialization (random and AE), the best results on

Dataset →	SVHN		CIFAR-10		CIFAR-100	
Method ↓ / Noise rate →	20%	60%	20%	60%	20%	60%
CE	82.68±0.34	42.22±0.41	76.31±0.16	37.96±0.24	47.93±0.55	20.38±0.75
GCE [9]	93.46±0.10	57.53±0.93	88.09±0.40	64.08±1.29	61.26±0.14	48.27±0.55
APL [10]	95.08±0.06	69.99±0.59	89.17±0.30	80.18±0.63	**62.35±0.57**	**48.28±1.35**
Small Loss	90.84±0.39	74.56±0.43	86.14±0.10	77.81±0.10	59.49±0.23	47.97±0.66
k-NN	95.38±0.04	76.55±1.48	89.17±0.17	80.68±0.35	59.77±0.28	45.42±0.38
TE-kNN	**95.48±0.13**	**77.04±1.56**	**89.37±0.13**	**81.55±0.42**	60.29±0.39	45.56±0.19
APL (AE init.)	95.12±0.07	71.85±0.50	88.74±0.33	79.76±0.60	**64.55±0.64**	41.40±1.17
k-NN (AE init.)	95.67±0.04	77.99±0.97	89.99±0.15	82.09±0.24	61.20±0.19	47.74±0.58
TE-kNN (AE init.)	**95.79±0.06**	**78.71±1.28**	**90.24±0.40**	**83.20±0.10**	61.69±0.32	**48.36±0.40**

Table 1: Means and standard deviations of the test accuracies over 3 runs. Best results for each type of initialization are shown in **bold**.

the SVHN and CIFAR-10 datasets are obtained by TE-kNN, which slightly outperforms the classical k-NN. The use of AE initialization is beneficial for k-NN and TE-kNN on all datasets. On CIFAR-100 with 20% noise TE-kNN is outperformed by APL and GCE, while on 60% label noise our approach is also surpassed by the small loss method. However, when using AE initialization, TE-kNN outperforms both k-NN and APL on CIFAR-100 in the 60% noise setting, while on 20% noise it is surpassed only by APL. In order to analyse the impact of k on the performance, we show in Figure 1a the test set accuracy obtained by k-NN and TE-kNN on the CIFAR-100 dataset using different values of k. TE-kNN is more robust to the choice of k than k-NN and obtains good performance even for a very small number of neighbours due to the temporal ensembling strategy. However, when using larger k values, the differences between the two approaches are smaller. Figure 1b illustrates the evolution of the sample selection process for TE-kNN. It shows the proportion of samples that are correctly identified as being clean from the samples selected by TE-kNN for each epoch on CIFAR-10 using 60% label noise. The selection of clean samples improves during training.

6 Conclusions

This paper proposed a k-NN inspired sample selection method based on a temporal ensembling strategy. We provided proof-of-concept results showing the promising performance obtained by our approach on synthetic symmetric label noise. Future work will focus on evaluating TE-kNN using more challenging datasets and real-world label noise. Additionally, extensions of the k-NN strategy using class-balancing techniques and adaptive thresholds for the scores such as the ones from [5, 6] will be envisaged.

References

[1] Ahmet Iscen, Jack Valmadre, Anurag Arnab, and Cordelia Schmid. Learning with neighbor consistency for noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4672–4681, 2022.

[2] Youze Xu, Yan Yan, Jing-Hao Xue, Yang Lu, and Hanzi Wang. Small-vote sample selection for label-noise learning. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*, pages 729–744. Springer, 2021.

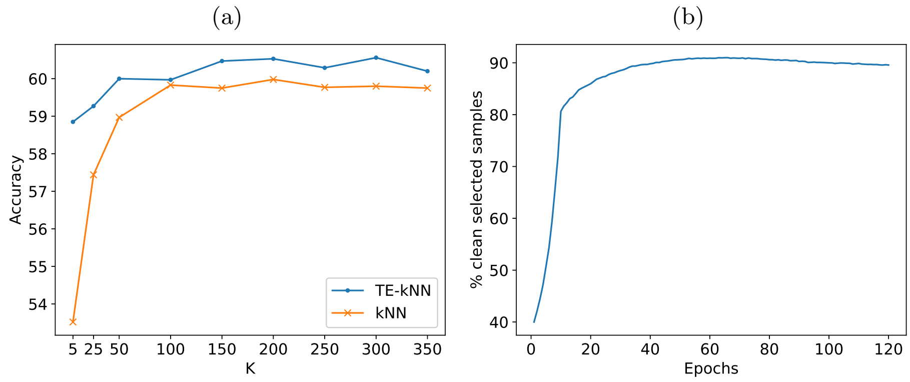

Fig. 1: Test set accuracy using different values of k for kNN and TE-kNN on the CIFAR-100 dataset using 20% noise rate (a) and proportion of samples that are correctly identified as clean by TE-kNN during training on CIFAR-10 (b).

[3] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[4] Dara Bahri, Heinrich Jiang, and Maya Gupta. Deep k-nn for noisy labels. In *International Conference on Machine Learning*, pages 540–550. PMLR, 2020.

[5] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-objective interpolation training for robustness to label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6606–6615, 2021.

[6] Chen Feng, Georgios Tzimiropoulos, and Ioannis Patras. Ssr: An efficient and robust framework for learning with unknown label noise. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022.

[7] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972.

[8] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[9] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[10] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning*, pages 6543–6553. PMLR, 2020.

[11] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

[12] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[13] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.