Revisiting 'Symmetric Cross Entropy for Robust Learning with Noisy Labels': A critical review

Daniel Wass **Martin Köling** **Angus Hansson** **Patrik Ekman**

Abstract

Deep Neural Networks (DNNs) for classification require large labelled data sets. Annotating data is a costly and error-prone cumbersome process that may require specific domain expertise or resource-demanding computational tasks. In this paper, we evaluate the performance of the Symmetric Cross Entropy (SCE) loss function on data sets with noisy labels. Different types of noise of different ratios are added to the data, and the performance of SCE is compared to the most commonly used loss function Cross Entropy (CE). In order to gain a better understanding of the negative effects of label noise and how well SCE can handle this problem, we have conducted a series of experiments. The results show that SCE is superior over CE on noisy data. Even though the approach appears robust, noise that imitates authentic noise, appears to be a tough nut to crack.

1 Introduction

DNNs in general, and Convolutional Neural Networks (CNNs) in specific, have become major popular choices for solving image classification problems (Karimi et al., 2019; Han et al., 2019). Today, CNNs are used in numerous everyday applications, such as Facebook's and Instagram's face recognition functions (Shepley, 2019). The ability to identify and extract features, together with high performance, is often shown superior to other machine learning techniques (de la O Arevalo, 2019). However, CNNs demand a vast amount of training data to perform on a high level (Karimi et al., 2019). To obtain such data sets, manual or automated annotation of collected data is necessary, a time consuming and costly process (Wang et al., 2019). Also, annotating real-world data sets almost inevitably results in high degrees of mislabelled data points, also called noisy labels. CNNs trained on data sets with noisy labels often show a significant reduction in performance, making noisy labels a key issue to handle when applying the neural network model to real-life problems (Han et al., 2019; Tang and Eliasmith, 2010).

The objective for this study has been to critically review one of the most recently presented techniques for training DNNs with data sets containing noisy labels. In the paper 'Symmetric Cross Entropy for Robust Learning with Noisy Labels', Wang et al. (2019) present the approach of symmetric cross entropy learning (SL), introducing a symmetric cross entropy (SCE) loss function, inspired by the symmetric Kullback-Leibler (KL) divergence and most commonly used loss function cross entropy (CE), to manage mislabelled data points. The authors apply different loss functions to an eight-layer CNN, including other approaches to the issue and the common CE, and compare their performance to the SCE loss on a few benchmark data sets, where the CIFAR-10 by Krizhevsky et al. (2009) is the most discussed. To investigate the loss functions' behaviour on noisy data sets, Wang et al. (2019) add noise of different ratios and types, i.e. symmetric and asymmetric noise, of ratios $\eta_{sym} \in [0.0, 0.8]$ and $\eta_{asym} \in [0.0, 0.4]$ respectively. They claim that DNN implementing the CE loss suffer from a class bias, meaning high differences between classes "easy" to learn and "hard" to learn. On noisy data sets, the bias is amplified, the authors state, leading to *overfitting* of easy labels, and *under-learning* of hard classes. The paper presents evidence that by implementing a SCE loss function, CNNs can become more robust to noisy data and the problems of overfitting and

Project report for DD2424

under-learning (Wang et al., 2019). In this paper, we implement two eight-layer CNNs, one using CE and the other SCE. We evaluate the models on data with different levels of noise, in an attempt to reproduce, but also critically examine, the results of the original paper. By providing further evidence of the robustness of the SCE loss function, this study contributes with validation of the superiority of SCE over CE. However, some critique is directed towards parts of the original paper.

2 Related Work

As the usage of DNN has increased the interest for managing noisy labels, a key issue, has grown. Numerous studies on how to handle mislabelled data have been published.

A common selection of methods, that can be summarised as Label Correction Methods, attempt to early identify the mislabelled data and correct them (Nicholson et al., 2015). This can be done by polishing the labels, where the data is run through multiple models to determine the correct label. Each sub-model tries to interpret the correct ground-truth label, and the label with a majority vote is then chosen (Teng, 1999). Another approach is to use self-training correction. An algorithm first filters and divides the original data set into one clean and one noisy subset. The model then uses the clean set to re-interpret the probability of a noisy data point belonging to a specific label. The label with the highest probability is then assigned to the data point. Lastly the datasets are combined for training the model. These models are often computationally heavy and requires knowledge of correct labelled data.

Another selection of models can be described as loss function methods. This is the type where Wang et al. (2019) approach belongs. Instead of identifying and changing mislabelled data, these methods attempt to reduce the impact of noisy labels by implementing a loss function less affected by it. The mean-absolute-error (MAE) is such a loss function, with a proved higher robustness and tolerance for noisy data, compared to the often used cross-entropy-loss (CE), in which the trade off is a much slower computational process with MAE compared to CE (Ghosh et al., 2017). Comparisons to CE is also implemented in the experiments of this study. A common issue is the demand for prior knowledge of the severity of noisy labels in a dataset. Many of the methods managing a loss function, including Wang et al. (2019), requires tuning of a hyperparameter determining the degree of mislabelled data. This is the alleged advantage of the recent findings of the Peer Loss Function approach (Liu and Guo, 2020), that enables efficient training over mislabelled data without knowledge of the proportional noise.

The symmetric learning technique proposed by Wang et al. (2019), also the foundation of this study, is based upon the CE loss function and the symmetric KL-divergence. The lack of robustness when using an CE for noisy labelled data, is confirmed in other studies. Zhang and Sabuncu (2018) writes that the usage of CE can lead to a class-biased model, where labels difficult to predict are emphasised. This is not a problem when training a model on clean data, but can lead to overfitting issues when training on noisy data. Similar results are presented in Yi and Wu (2019), where CE shows overfitting tendencies compared to other loss functions.

3 Data

The data used in the experiments of this paper has been the CIFAR-10 data set, from Krizhevsky et al. (2009). It consists of 60,000 RGB colour images with size $32x32$, and was loaded directly from the Tensorflow data set library. The data was divided according to the original paper, i.e. 50,000 of the images were used for training, and 10,000 for testing. The images belong to 10 different classes, corresponding to what the image illustrates, i.e., the labels are, indexed 0 to 9: (0) airplane, (1) automobile, (2) bird, (3) cat, (4) deer, (5) dog, (6) frog, (7) horse, (8) ship, and (9) truck. This was one of the data set used by Wang et al. (2019), but it is also often used as a benchmark data set in deep learning, which we have experienced during the course of *Deep Learning in Data Science* at KTH.

3.1 Data preparation

The CIFAR-10 data set is user friendly and the necessary pre-processes were few and uncomplicated. The RGB-colour coding of the images was normalised to be between zero and one, i.e. divided by

255. Furthermore, the training data was standardised to have zero mean and unit variance, and the testing data was normalised by the same norms, i.e. by the training mean and standard deviation.

Three different types of noise was added to the data sets, i.e. symmetric and asymmetric noise, defined according to the original paper, and an additional asymmetric one, here called *extended asymmetric noise*. The types of noise were added one at a time, i.e. no combinations of the noise were applied.

3.1.1 Symmetric Noise

Symmetric noise was added to all classes uniformly, by randomly changing the labels of the amount of data points corresponding to the noise rate. This essentially means that the total noise across the entire training data corresponded to the noise rate.

3.1.2 Asymmetric Noise

Asymmetric noise was generated by changing the labels of specific source classes to that of specific target classes, according to the amount corresponding to the noise rate. The source and target classes are ones that can be considered similar, and thus can be expected to be mixed up in data sets with authentic noise, e.g. a flying bird could perhaps be mistaken for an aircraft, but not for a horse.

The asymmetric type of noise of this paper was added according to the same scheme as in the original paper, i.e. the labels were changed according to the following scheme: TRUCK (9) $\rightarrow$ AUTOMOBILE (1), BIRD (2) $\rightarrow$ AIRPLANE (0), DEER (4) $\rightarrow$ HORSE (7), CAT (3)$\leftrightarrow$ DOG (5) (Wang et al., 2019). Note that, as noise was not added to all classes, but merely five (the source classes), the *total* noise ratio of this type was lower than in the case of symmetric noise.

3.1.3 Extended asymmetric noise

One can expect, that in a data set with authentic noise, not only can birds be mislabelled as aircraft, but also aircraft as birds. Thus, we extended the asymmetric noise of the original paper, by making the asymmetric noise two-way. Accordingly, the scheme for changing lables was: TRUCK (9) $\leftrightarrow$ AUTOMOBILE (1), BIRD (2) $\leftrightarrow$ AIRPLANE (0), DEER (4) $\leftrightarrow$ HORSE (7), CAT (3)$\leftrightarrow$ DOG (5). Note that also this type of noise leads to a lower total noise ratio than for the symmetric one, but a higher than for the original asymmetric one.

4 Methods

This section initially provides the reader with an explanation of the experimental setup, specifying the DNN architecture and hyper-parameters used in the investigative experiments of this study. Following this is an intuitive theoretical introduction of the two different loss functions being compared, CE and SCE, highlighting important differences in how they handle the problem of noisy data. Lastly, a methodological discussion on delimitations is performed, discussing and arguing for some of the choices made in this study.

4.1 Experimental setup

As in Wang et al. (2019), two different DNN models were applied, one with regular CE, and the other with SCE. Both models consisted of six convolutional layers followed by one fully connected layer and finally the output layer. Just as Wang et al. (2019), Tensorflow was used, and more specifically the Keras API, adding layers sequentially using max-pooling between every second convolutional layer. Further, the ReLU activation function was used for all layers except the output layer, where the Softmax function was applied, also in accordance to the original paper. Regularisation was applied to the seventh layer (the first fully connected) and batch normalisation was applied on each layer. The choice of using batch normalisation was considered crucial to eliminate the problem of vanishing gradients in the learning process. He-initialisation was used to initialise the kernel weights matrices. The optimisation algorithm used was Stochastic Gradient Descent, with a batch size of 128, a learning rate decay of $5 * 10^{-3}$ and a momentum of 0.9. α was set to 0.1, β to 1.0, and A to -4. The learning rate was initially set to 0.1, and then divided by 10 after 40 and 80 epochs, with a total of 120 epochs. After each epoch, the data was shuffled. The regularisation coefficient λ was set to 0.01, which is the

same as in the source code of the original study. The performance metrics used for model evaluation was the class-wise and overall accuracy.

4.2 Cross entropy

The Cross Entropy loss function for a sample can be written as

$$l_{ce} = -\sum_{k=1}^{K} q(k|\mathbf{x}) \log(p(k|\mathbf{x})) \tag{1}$$

where $p = p(k|x)$ is the probability distribution of a sample belonging to a class k and $q = q(k|x)$ represents the ground-truth probability distribution.

Some of the weaknesses with CE on noisy data is mentioned in the *Previous Work*-section of this paper. These weaknesses are also emphasised by Wang et al. (2019) and later confirmed by the initial experiments made in this study. We found that in an early stage of training (10 epochs), CE learning was class-biased for both clean labels and with 40% symmetric noise. That is, the computed test accuracy per class varied to a great extent. As training proceeded on the clean data, the model were able to increase the accuracy and decrease the variation between the classes. However, the model still struggled to achieve uniform accuracy among classes, when trained on the noisy data.

In summary the weakness of implementing CE on noisy data is a decrease in overall test accuracy and a class-bias behaviour with high accuracy variance between the labels.

4.3 Symmetric cross entropy

To overcome the issues with CE and noisy labels, Wang et al. (2019) suggest the approach of SL, using the loss function SCE. SL is an approach where the alleged ground-truth probability distribution q is not seen to solely represent the truth, due to the knowledge that some labels are actually noisy. Instead, the trained probability distribution p is to some extent considered to be another valid representation of the truth.

The departure point of SCE is that of the symmetric KL-divergence. The KL-divergence $KL(q||p)$ is, from a classification point-of-view, to learn the classification model f how to compute a probability distribution $p = p(k|x)$ similar to the ground-truth distribution $q = q(k|x)$. This is equivalent to minimising $KL(q||p)$ with respect to these two distributions:

$$KL(q||p) = H(q, p) - H(q) \tag{2}$$

In a situation with noisy data however, as q can not be considered to be an entirely proper representation of the ground-truth, and p being another acknowledged representation of the truth, the reverse KL-divergence $KL(p||q)$ must be also be taken into account. When combining the KL-divergence $KL(q||p)$ and its reversed version $KL(p||q)$, one arrives at the symmetric KL-divergence, that thus can be expressed as:

$$SKL = KL(p||q) + KL(q||p) \tag{3}$$

Taking the idea of complementing the initial ground truth with that of another possible representation to the concept of entropy, one arrives at the SCE, complementing the CE with its reversed version, the reverse cross entropy (RCE):

$$SCE = CE + RCE = H(q, p) + H(p, q) \tag{4}$$

The corresponding loss function of RCE, i.e. the reverse of the CE loss function is thus:

$$l_{rce} = -\sum_{k=1}^{K} p(k|\mathbf{x}) \log(q(k|\mathbf{x})) \tag{5}$$

The SCE loss is then defined as the sum of the two contradictory loss functions:

$$l_{sce} = l_{ce} + l_{rce} \tag{6}$$

To further increase the robustness of SCE Wang et al. (2019) propose an implementation of two decoupled hyper-parameters; α and β. The function of α is to monitor CE and its overfitting behaviour, while β is implemented for monitoring the overall robustness of RCE. The final and formal expression of SL is therefore:

$$l_{sl} = \alpha l_{ce} + \beta l_{rce} \tag{7}$$

We implemented the SCE loss function manually, whereas the CE loss function was imported from the Keras API.

4.4 Methodological discussion

The approach used by Wang et al. (2019) to reduce the influence of noise on the performance of a DNN, was to implement the SCE loss function, explained closer in section 4.3. This study took the same approach to investigate if, and to what extent, SCE changes model performance on noisy data, specifically compared to CE, as well as to confirm or discard some of the results in the original study by Wang et al. (2019). The methods and hyper-parameters used in this study were thus almost identical to those of the original study, however with a few side tracks investigated. Further on this note, and before proceeding with more methodological discussions and delimitations, it should be highlighted that the purpose of this study was not to optimise the performance of the implemented classification models, but rather to serve as an introductory examination of how noisy data can be handled practically, with exclusive focus on SCE, comparing the performance of SCE to that of CE in cases with noisy data. Hence, because the comparison aspect of this study was considered more central than the actual metric performance in percentages, hyper-parameter tuning as such was not considered as important as it normally should when optimising a model. This was yet another reason for mainly implementing the same hyper-parameters used and optimised by Wang et al. (2019). However, during the experiments, some questionable characteristics of the learning processes and the choices of noise and hyper-parameters by Wang et al. (2019) were found, two examples being the issue of asymmetric noise discussed in section 3.1.3, as well as the amount of regularisation used, i.e. the regularisation term λ. In such cases the corresponding variables and hyper-parameters were altered to objectively ensure that the performance differences did in fact not depend on such specific choices.

SCE in comparison to CE was the chosen approach and a delimitation compared to the more exhaustive loss function comparisons performed by Wang et al. (2019), mainly because CE has been the loss function used during the practical parts of the course in which this project has been performed; *Deep Learning in Data Science* at KTH, as well as with arguments presented by Wang et al. (2019), emphasising that CE is the most commonly used loss function. Moreover, CIFAR10 was used as it is the most frequently discussed data set in Wang et al. (2019).

Both models, with SCE and CE loss functions, were tested on clean data and on data with the three types of noise of different rates. In Wang et al. (2019), symmetric noise rates of 0, 20, 40, 60 and 80 percent were used. In this study however, we argue that noise levels of over 50 percent are too high to reliably imitate authentic noise, in which the results of a classification model would be of actual use, and in such cases, adjusting the model would be of less importance than adjusting the actual labelling of the data. Therefore, the only noise levels chosen for this study, both for symmetric and asymmetric noise, were 0, 20 and 40 percent.

Another difference of this study compared to Wang et al. (2019), is that the tests were ran merely once with a random seed, rather than averaging the results of five random runs. The random seed ensured that the preconditions for the models were always the same, thus making the comparison objectively fair, which is also the main reason for why it was done. While an average from multiple runs in some sense prove that one model outperforms the other over time, rather than in a single case, one could also argue that if SCE actually do perform better than CE, it should do so independently of the preconditions, and that a setting with equal preconditions thus provide another important aspect in the comparison. However, to ensure that this choice was not of great importance, sample tests of the models without random seed and with other random seeds were performed. The results from these sample tests did not prove any noticeable differences to the ones presented in the experiment section of this study.

5 Experiments

5.1 Results

The test scores can be seen, together with the scores of Wang et al. (2019), in *Table 1.* The overall accuracy of the models of the original study was significantly higher than the scores of our study for all data setups that are used in both studies. Continuing in terms of overall accuracy, the SCE loss was superior to the CE for all data setups for both studies, however only slightly for the clean data set in the original paper. Furthermore, regarding the results of this study, the difference in overall accuracy between SCE and CE increased (in favour of the SCE score) as the noise ratio increased when symmetric noise was added, whereas when adding asymmetric noise, the other way around, the difference decreased as the noise ratio increased. Following the complete test results of Wang et al. (2019), the same behaviour can be seen for symmetric noise, while for the asymmetric noise, the difference is rather stable.

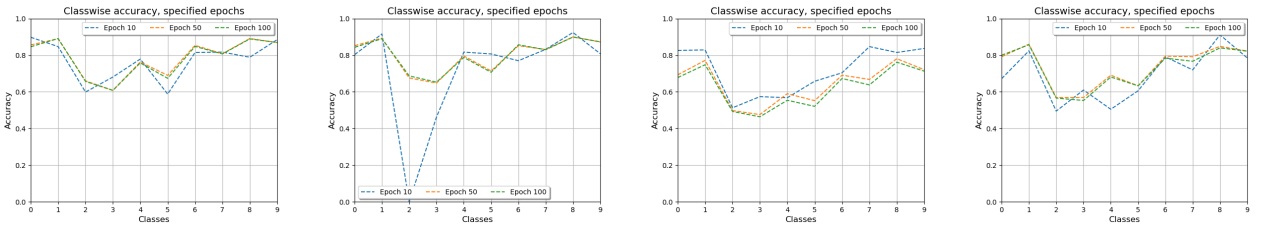
Table 1: The overall accuracy scores of this study and of Wang et al. (2019). The difference Δ is presented in percentage points.

			Accuracy			
	Data		*Our study*		*Wang et al.*	
Model	*Noise*	*Noise ratio*	Score (%)	Δ (pts)	Score (%)	Δ (pts)
Cross entropy	No noise	-	78.52	3.01	89.26	0.02
Symmetric learning			81.53		89.28	
Cross entropy	Symmetric	0.2	71.51	5.36	82.96	4.67
Symmetric learning			76.87		87.63	
Cross entropy		0.4	62.55	10.37	78.70	6.64
Symmetric learning			72.92		85.34	
Cross entropy	Asymmetric	0.2	73.00	3.16	85.98	2.26
Symmetric learning			76.16		88.24	
Cross entropy		0.4	65.96	1.52	78.51	2.13
Symmetric learning			67.48		80.64	
Cross entropy	Extended asymmetric	0.2	71.25	4.21		
Symmetric learning			75.46			
Cross entropy		0.4	62.71	5.18		
Symmetric learning			67.89			

5.2 Discussion

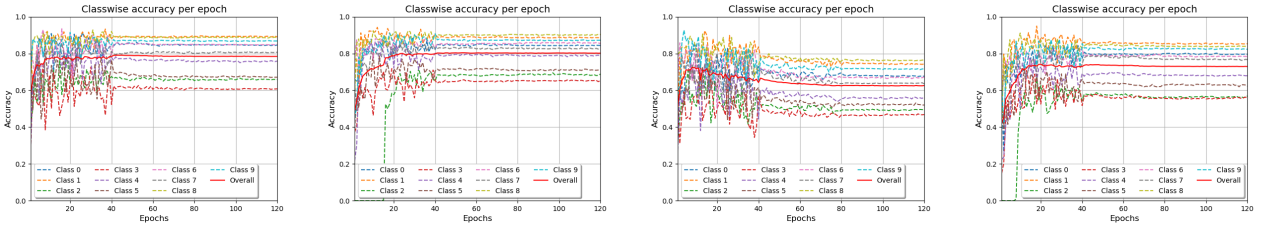
That the overall accuracy scores of our models are so much lower than the scores of Wang et al. (2019) is perhaps the first thing that comes to mind when observing *Table 1.* One reason to this could be the mentioned, but unspecified, data augmentation that Wang et al. (2019) performed. However, as boosting performance in general is out of scope of this study, we have simply accepted this difference, and instead focused on the performance differences of the two loss functions on noisy data.

The results show that the SCE loss function is more robust than CE when trained on manually added noise, and its superiority is especially significant on the symmetric noise. The first was expected since the added ℓ_{rce} regularises the noisy labels, and as the scalar ratio β/α was as high as 10 in the tests, the trust in the true labels was such low compared to the trust in the learned characteristics, that the model was forced not to overfit on the noise. The latter tendency is less clear, but remembering that the total noise ratio is lower for asymmetric and extended asymmetric noise, the smaller difference between the models can perhaps partly be explained - the CE-model's performance decrease ought to be scaled faster with the decrease in total noise than the SCE-model's. Furthermore, the mistrust in the labelled class $q(k_i|\mathbf{x}) = 1$ inevitably comes with increased trust in the most probable class k_p. If k_i is mislabelled, but easily separable from the most probable class, the first term of *Eq. 7* is overridden by the second term, clearly stating that x does not belong to k_i, and thus the entropy indicates towards the correct class (or at least not towards the mislabelled one). For the random symmetric noise this should, for most classes occur around eight out of nine times, e.g. an image of the class 'bird' should in most cases be easily separable from all other classes except 'airplane'. For the asymmetric noise, the labels are always switched to a label of a similar class, and thus in all cases,



(a) Clean data, CE (b) Clean data, SCE (c) 40% symmetric noise, CE (d) 40% symmetric noise, SCE

Figure 1: Class-wise accuracy after epoch 10 (blue), 50 (orange) and 100 (green).



(a) Clean data, CE (b) Clean data, SCE (c) 40% symmetric noise, CE (d) 40% symmetric noise, SCE

Figure 2: Class-wise (dashed lines) and overall (bold red line) accuracy after each epoch.

the 'bird' is mislabelled into an 'airplane'. As $p(\text{'bird'}|x)$ and $p(\text{'airplane'}|x)$ are close to each other more often than in the case of symmetric noise, and as the first term of *Eq. 7* further decrease the entropy for classifying x as an 'airplane', the images that really show a bird, are more often falsely classified as an 'airplane' when the asymmetric noise is added to the data. This concerns, of course, all the changed source classes (not only 'bird'), and also the extended asymmetric noise, and is part of the explanation to why the SCE have shown to be less robust and less superior to CE in cases of asymmetric noise, both in our study and the original study.

One of the main weaknesses of CE, as presented by Wang et al. (2019), is that it seems to exhibit overfitting on easy classes and under-learning on harder classes, a characteristic being extra amplified when some of the training labels are noisy. In particular, they claim that the under-learning on harder classes is a barrier to a higher overall accuracy, and that the extra term that SCE provides CE with makes the model more tolerant to noise, promoting more sufficient learning of harder classes as well as improving the robustness of the model - i.e. decreasing the accuracy gap between easier and harder classes (Wang et al., 2019). *Figure 1* demonstrates the class-wise accuracy reached after 10, 50 and 100 epochs in the experiments of our study, comparing CE to SCE with clean data (subfigures a and b) as well as with 40% symmetric noise (subfigures c and d), while *Figure 2* likewise demonstrates the class-wise and overall accuracy for each epoch. Analysing the figures, it is clear that the lower accuracy levels (harder classes) are actually pushed up when using SCE rather than CE. It is also clear that the higher levels (easier classes) are either pushed up or remain at the same level. These findings indicate that our experiments demonstrate the same tendencies as those of Wang et al. (2019), although not as evident. In fact, while SCE actually seems to reduce the issue of under-learning, it does not evidently remove the issue of over-learning, leading to questionable results of whether the accuracy gap between lower and higher levels is actually decreasing. However, comparing model performance in *Table 1*, the more sufficient learning of harder classes could potentially be a factor facilitating and raising the lower accuracy barrier, leading to a higher overall accuracy.

The previously mentioned tendency of overfitting when using CE is confirmed by our experiments, clearly illustrated in *Figure 2c*, as well as in (Wang et al., 2019, Figure 5a). After less than ten epochs a ten percent continuous decrease in overall accuracy is shown. Our analysis of this behaviour is that the model overfits and adapts to the mislabelled data, which leads to a decreasing performance on the clean labelled test data. In the figures presented by Wang et al. (2019) the accuracy does not begin to decline until epoch 40, this is probably due to the higher learning rate implemented in our model than in the corresponding case of Wang et al. (2019). Our initial thought to counter this problem

was to try different levels of regularisation, i.e. changing the regularisation coefficient λ. Therefore, two complementing tests with lambda equal to 0.001 and 0.1 were performed. However, the results demonstrated no noticeable differences. Further, looking at *Figure 2d*, SCE actually does seem to reduce this tendency.

6 Conclusion

The objective of this paper was to critically review the approach of SL, as proposed by Wang et al. (2019). We created a convolutional neural network and implemented the topical loss function, followed by a reproduction of some of the experiments of the original paper to verify its alleged performance. Our results imply that implementing the SCE is superior over the CE, however an even more generalised study is required to fully support this claim. Furthermore, we have found that SCE facilitates more sufficient learning of harder classes, and equal or better learning of easier classes, which are the reasons for the overall higher accuracy. However, it is not evident whether SCE actually decreases the accuracy gap between easier and harder classes. Even though the study, in line with Wang et al. (2019), shows that SL is a robust approach to the problem of noisy labels, the authors want to highlight its difficulties in handling asymmetric noise. Such noise can be considered more likely in authentically mislabelled data sets, which is the fundamental problem formulation of this as well as the original study. An interesting approach to further research would thus be to more deeply investigate the dynamics of DNNs on data sets with asymmetric, or even authentic, noise.

References

de la O Arevalo, L. (2019). Comparison of image classification algorithms using meibography images*.

Ghosh, A., Kumar, H., and Sastry, P. S. (2017). Robust loss functions under label noise for deep neural networks. page 1919–1925.

Han, J., Luo, P., and Wang, X. (2019). Deep self-learning from noisy labels.

Karimi, D., Dou, H., Warfield, S. K., and Gholipour, A. (2019). Deep learning with noisy labels: exploring techniques and remedies in medical image analysis.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Liu, Y. and Guo, H. (2020). Peer loss functions: Learning from noisy labels without knowing noise rates.

Nicholson, B., Zhang, J., Sheng, V. S., and Wang, Z. (2015). Label noise correction methods. pages 1–9.

Shepley, A. J. (2019). Deep learning for face recognition: A critical analysis. *ArXiv*, abs/1907.12739.

Tang, Y. and Eliasmith, C. (2010). Deep networks for robust visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1055–1062. Citeseer.

Teng, C.-M. (1999). Correcting noisy data. page 239–248.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 322–330.

Yi, K. and Wu, J. (2019). Probabilistic end-to-end noise correction for learning with noisy labels. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Z. and Sabuncu, M. R. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels.