Hybrid Loss for Improving Classification Performance with Unbalanced Data

Thanawat Lodkaew and Kitsuchart Pasupa(✉)

Faculty of Information Technology, King Mongkut's Institute of Technology
Ladkrabang, Bangkok 10520, Thailand
lodkaew.thanawat@gmail.com, kitsuchart@it.kmitl.ac.th

Abstract. Unbalanced data is widespread in practice and presents challenges which have been widely studied in classical machine learning. A classification algorithm trained with unbalanced data is likely to be biased towards the majority class and thus show inferior performance on the minority class. To improve the performance of deep neural network (DNN) models on poorly balanced data, we hybridized two well-performing loss functions, specially designed for learning imbalanced data, mean false error and focal loss. Since mean false error can effectively balance between majority and minority classes and focal loss can reduce the contribution of unnecessary samples, which are usually samples from the majority class, which may cause a DNN model to be biased towards the majority class when learning. We show that hybridizing the two losses can improve the classification performance of the model. Our hybrid loss function was tested with unbalanced data sets, extracted from CIFAR-100 and IMDB review datasets, and showed that, overall, it performed better than mean false error or focal loss.

Keywords: Class imbalance · Deep neural network · Loss function

1 Introduction

Class imbalance occurs when the samples of each class are not equally represented, *i.e.* the numbers of representatives differ widely: many real-world datasets show this imbalance [8,13,17,18]. Since this is extremely common in practice, it has been widely studied in classical machine learning. Commonly, there are two types of imbalance—long-tailed imbalance [15] and step imbalance [3]. In step imbalance, classes are grouped into majority and minority classes. The two classes have different numbers of samples, but the number of samples is equal within majority classes and equal within minority classes. For long-tailed imbalance, the class frequency distribution is long-tailed, the samples of a few classes occupy most of the data, while samples of most classes rarely appear. In binary classification, when a dataset is imbalanced, it is a step imbalance. This paper focuses on binary classification.

Recently, deep neural networks (DNNs) have been used for various classification tasks, *e.g.* image and text classification, and they have achieved excellent

H. Yang et al. (Eds.): ICONIP 2020, CCIS 1332, pp. 807–814, 2020.
https://doi.org/10.1007/978-3-030-63820-7_92

performance. However, DNNs perform poorly on imbalanced data due to ineffective learning [3,6]. In binary classification, when classification algorithms based on DNNs are trained with unbalanced data, classifiers will prefer the negative (majority) class and achieve high accuracy on it. However, it will show lower accuracy on the positive (minority) class.

Existing methods use two strategies for dealing with imbalanced data [9]—data sampling and algorithmic adjusting. There are two data sampling techniques—over-sampling the positive class and under-sampling the negative class. However, each techniques has disadvantages: over-sampling can easily cause model over-fitting, due to repeatedly duplicated samples, whereas under-sampling may throw away valuable information, and it is not practicable for extremely unbalanced data. Algorithmic adjusting changes the learning process, so that it can give higher importance to the positive class. One technique for adjusting the algorithm is cost-sensitive learning, which considers the misclassification costs [19]. If it is applied to a DNN model, the learning will jointly optimize the network parameters and misclassification costs, instead of optimizing the network parameters alone [10,21]. It will be difficult to simultaneously optimize the network parameters and misclassification costs, when the imbalance is large [7]. However, recent work has addressed the class imbalance problem, without adding additional parameters [14,22]. Solutions proposed in [14,22] allow the model to optimize just the network parameters. To clarify, they tried to solve the problem by modifying just the existing loss functions and did not alter the models. It was quite simple but effective. The essential advantage of this strategy for solving the problem is that it is easy to implement and use with existing DNN models.

Here, we studied two well-performing loss functions, namely mean false error (MFE) [22] and focal loss [14], specially designed to combat the imbalance problem. These two loss functions used different perspectives to make learning the model concentrate more on the positive class. Focal loss differentiates between easy samples (samples with low losses) and hard samples (samples with high losses), so that it can lower the weight of the loss contribution of easy samples and focus training on hard samples. This gives more importance to the positive class, because most easy samples are in the negative class. The mean false error technique changes the total error by summing the negative and positive sample errors separately. This effectively balances between the loss contributions of both classes and allows the positive class to have a substantial contribution in calculating the total loss.

There is a drawback for each loss. For focal loss, the contribution of the negative class (or easy samples class) to the total loss is reduced. However, the total loss is an average over the whole data, so losses from negative samples can still dominate it. For mean false error, although the total loss is calculated by summing the average losses of both classes, the loss from the negative class can still dominate the overall loss, because of the effect of the easy samples. Moreover, mean false error will work best, if every batch of training data contains at least one positive sample. If there is no positive sample in a batch, the total loss will be biased by the average of negative class, *i.e.* the easy samples.

To avoid the drawbacks, inspired by these two approaches, we formed a hybrid solution and defined a new loss function—the hybrid loss—so that advantages of each loss will compensate for the drawbacks of the other.

Our main contributions are: Firstly, we explored the ideas behind the mean false error and focal loss ideas, to understand how they perform, when the data is unbalanced. Secondly, we defined a hybrid loss function, a hybrid of mean false error and focal loss solutions, which combines advantages of the two ideas, and we showed that the two loss functions can be combined in an efficient way. Lastly, we tested our hybrid function with image and text datasets. For each dataset, a variety of imbalance levels was applied.

2 Related Works

2.1 Imbalanced Learning

Anand *et al.* [2] studied the effect of class imbalance and found that it adversely affects the backpropagation algorithm. The loss of the negative class rapidly decreased, whereas the positive class loss significantly increased in early iterations and the network would often converge very slowly. This occurred because the negative class completely dominated the network gradient used to update the weights. To deal with this, we need to increase the positive class contribution and correspondingly decrease the negative class contribution.

2.2 Focal Loss

Focal loss, $FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t)$, was a modification of cross entropy loss [14]. A modulating factor $(1-p)^\gamma$ was added to the cross entropy loss. For notational convenience, let p is the predicted probability and y is the ground-truth class. p_t will be p if $y = 1$ and be $1 - p$ for otherwise. By the equation of focal loss, $\gamma \geq 0$ is a tunable focusing parameter. In practice, α_t will be α, if the ground-truth class of sample is the positive class and be $1 - \alpha$ for otherwise.

The motivation for defining the focal loss is that cross entropy loss is not able to correctly balance the weights of positive and negative samples due to the imbalance. Although adding a weighting factor α partially addresses the problem, it cannot differentiate between easy samples and hard samples. Usually, most of easy samples are from negative class, and they hugely contribute to the total loss and dominate the network gradient. In general, hard samples add more discriminative information than easy samples [23], so that learning from hard samples is more effective than learning from easy ones. For this reason, the contribution of easy samples needs to be reduced while learning, so that the model can concentrate on learning hard samples.

Focal loss was designed to down-weight easy samples by adding a modulating factor to the cross entropy loss. This factor reduces the loss contribution from easy samples and focuses training on hard negative samples. Define

$l_{FL} = \frac{1}{n}\sum_{i=1}^{n} -\alpha_t^{(i)}(1-p_t^{(i)})^{\gamma}\log(p_t^{(i)})$, as a total loss form of an α-balanced variant of focal loss, where n is the number of samples.

We considered focal loss as a reference for our improved method, described in the next section.

2.3 Mean False Error

Mean false error was derived from a mean squared error (MSE) [22], by separating the calculation of the total MSE for all samples to a sum of an average losses of negative and positive samples separately: $l_{MFE} = l_{MSE_-} + l_{MSE_+}$, where $l_{MSE_-} = \frac{1}{n_-}\sum_{i=1}^{n_-}\frac{1}{2}(y^{(i)} - p^{(i)})^2$ and $l_{MSE_+} = \frac{1}{n_+}\sum_{i=1}^{n_+}\frac{1}{2}(y^{(i)} - p^{(i)})^2$. Based on the equations, $y^{(i)}$ is the ground-truth class of sample i and n_- and n_+ are the numbers of negative or positive samples.

The motivation for introducing mean false error is that a MSE is not able to capture losses from the positive class effectively. That is, loss contributions from negative samples will overrule the contribution from positive samples, due to the higher volume of negative samples. Thus it computes the total loss from a sum of separate calculations of the average loss of each class. This allows the positive class to more fully contribute to updating weights of the network. In experiments on various benchmark datasets, Wang *et al.* [22] showed that mean false error performed better than a simple MSE approach. They further improved mean false error with mean squared false error (MSFE) [22]. Both of these variations were compared with our hybrid method—see Sect. 5.

3 Our Method

The principal advantage of focal loss is that it can control the difference between easy and hard samples and increase the loss contribution of the positive class by reducing the importance of easy samples. A weighting factor was added to the loss to balance the contribution of positive and negative samples. However, since the total loss is an average for both positive and negative classes, the negative class can still dominate the total loss. The mean false error solution diminishes this effect, because it can make positive class more important during training.

We showed that the advantage of each loss can address the drawback of the other. Hence, to more effectively learn unbalanced data, we mimicked the mean false error total loss calculation, by summing average separately computed losses from both classes: $l_{Hybrid} = l_{FL_-} + l_{FL_+}$, where $l_{FL_-} = \frac{1}{n_-}\sum_{i=1}^{n_-} -\alpha_t^{(i)}(1-p_t^{(i)})^{\gamma}\log(p_t^{(i)})$ and $l_{FL_+} = \frac{1}{n_+}\sum_{i=1}^{n_+} -\alpha_t^{(i)}(1-p_t^{(i)})^{\gamma}\log(p_t^{(i)})$. l_{FL_-} and l_{FL_+} are the average losses of the negative and positive classes.

To use the hybrid loss in back-propagation algorithm, we need their derivatives. For focal loss, let $p = \sigma(x) = \frac{1}{1+e^{-x}}$, be an output of a logistic function, and x is an input of the logistic function. [14] define a quality $x_t = xy$. Based on the definition of p_t in Sect. 2.2, $p_t = \frac{1}{1+e^{xy}}$. Using p_t, the derivative for focal loss is: $\frac{\partial l_{FL}}{\partial x_t^{(i)}} = \frac{1}{n}\sum_{i=1}^{n} y^{(i)}(1-p_t^{(i)})^{\gamma}(\gamma p_t^{(i)}\log(p_t^{(i)}) + p_t^{(i)} - 1)$. For mean false error, the derivative is: $\frac{\partial l_{MFE}}{\partial x^{(i)}} = \frac{\partial l_{MSE_-}}{\partial x^{(i)}} + \frac{\partial l_{MSE_+}}{\partial x^{(i)}}$, where

$$\frac{\partial l_{MSE_-}}{\partial x^{(i)}} = -\frac{1}{n_-}\sum_{i=1}^{n_-}(y^{(i)} - p^{(i)})p^{(i)}(1-p^{(i)}), \tag{1}$$

$$\frac{\partial l_{MSE_+}}{\partial x^{(i)}} = -\frac{1}{n_+}\sum_{i=1}^{n_+}(y^{(i)} - p^{(i)})p^{(i)}(1-p^{(i)}). \tag{2}$$

Note that the derivative in (1) is used for the negative sample, while (2) is used for the positive sample.

Using the mean false error derivative, we can define the derivative for the hybrid loss by combining the derivatives of focal loss for negative and positive classes: $\frac{\partial l_{Hybrid}}{\partial x_t^{(i)}} = \frac{\partial l_{FL_-}}{\partial x_t^{(i)}} + \frac{\partial l_{FL_+}}{\partial x_t^{(i)}}$, where

$$\frac{\partial l_{FL_-}}{\partial x_t^{(i)}} = \frac{1}{n_-}\sum_{i=1}^{n_-} y^{(i)}(1-p_t^{(i)})^{\gamma}(\gamma p_t^{(i)}\log(p_t^{(i)}) + p_t^{(i)} - 1), \tag{3}$$

$$\frac{\partial l_{FL_+}}{\partial x_t^{(i)}} = \frac{1}{n_+}\sum_{i=1}^{n_+} y^{(i)}(1-p_t^{(i)})^{\gamma}(\gamma p_t^{(i)}\log(p_t^{(i)}) + p_t^{(i)} - 1). \tag{4}$$

As in mean false error, these derivatives are used for the corresponding samples from each class.

Our hypothesis is that our hybrid loss function will perform better than mean false error and focal loss, because it allows the positive class to contribute in its full extent to the total loss and differentiate between easy and hard samples at the same time.

4 Experimental Framework

4.1 Datasets

We use two benchmark datasets, CIFAR-100 [12] and IMDB review [16]. Originally, both datasets were balanced, but we extracted various imbalanced sets from them: (1) Unbalanced Sets from CIFAR-100: CIFAR-100 has 100 classes and contains 600 images per class, including 500 training and 100 testing images. For fair comparison, we created three different sets of data, labeled Household, Tree 1 and Tree 2, by following the setting of Wang *et al.* [22]. Each set of data had two classes and the representation of one class was reduced to three different imbalance levels, 20%, 10% and 5%. (2) Unbalanced Sets from IMDB Review: IMDB review is for binary sentiment classification: it contains 25,000 movie reviews for training and 25,000 for testing, and each set includes 12,500 positive and 12,500 negative reviews. We created three different sets of data by leaving 20%, 10% and 5% of positive reviews.

Table 1. Performance of ResNet-50 with different loss functions. The high F_1-score and AUC demonstrate that the loss function was suited for image classification on unbalanced data

Dataset	Imb. level (%)	Metrics	Method			
			MFE	MSFE	FL	Hybrid
Household	20	F_1-score	38.02 ± 0.03	40.15 ± 0.06	41.77 ± 0.02	**43.38 ± 0.04**
		AUC	73.58 ± 0.01	74.86 ± 0.02	75.00 ± 0.02	**75.24 ± 0.01**
	10	F_1-score	13.06 ± 0.07	13.31 ± 0.06	22.01 ± 0.02	**25.40 ± 0.05**
		AUC	60.78 ± 0.02	60.80 ± 0.01	60.61 ± 0.03	**65.24 ± 0.02**
	5	F_1-score	2.87 ± 0.04	6.99 ± 0.01	9.02 ± 0.05	**10.06 ± 0.03**
		AUC	51.55 ± 0.03	55.74 ± 0.03	**57.72 ± 0.03**	54.80 ± 0.04
Tree 1	20	F_1-score	38.86 ± 0.05	42.69 ± 0.08	34.12 ± 0.05	**50.12 ± 0.06**
		AUC	**80.09 ± 0.01**	79.60 ± 0.02	78.98 ± 0.01	79.62 ± 0.02
	10	F_1-score	36.63 ± 0.10	40.63 ± 0.12	33.33 ± 0.13	**42.49 ± 0.11**
		AUC	73.32 ± 0.03	74.82 ± 0.04	70.80 ± 0.02	**76.08 ± 0.03**
	5	F_1-score	32.38 ± 0.02	30.48 ± 0.02	26.67 ± 0.13	**33.33 ± 0.05**
		AUC	**80.68 ± 0.05**	79.96 ± 0.04	72.20 ± 0.03	80.08 ± 0.03
Tree 2	20	F_1-score	56.38 ± 0.05	57.71 ± 0.03	58.76 ± 0.03	**61.66 ± 0.05**
		AUC	82.22 ± 0.02	82.46 ± 0.02	81.92 ± 0.02	**82.53 ± 0.01**
	10	F_1-score	53.48 ± 0.06	57.10 ± 0.08	57.13 ± 0.03	**61.86 ± 0.08**
		AUC	80.90 ± 0.01	81.08 ± 0.03	79.58 ± 0.03	**81.30 ± 0.03**
	5	F_1-score	47.59 ± 0.07	43.59 ± 0.10	50.95 ± 0.09	**55.71 ± 0.03**
		AUC	72.04 ± 0.07	**78.88 ± 0.08**	65.52 ± 0.09	71.88 ± 0.07

4.2 Experiment Settings

Each unbalanced data set was split into training, validation and test sets. All three sets have the same imbalance ratio. As both CIFAR-100 and IMDR review, already had training and test sets, we chose 20% of samples from the training set for the validation set. The obtained training and validation sets are used for training model, and the test set is used for evaluating the trained model. The experiment was run five times with different random splits.

We used ResNet-50 [5] for image classification, and Transformer [20], that is represented in Keras document for sentiment classification. Both models used the Adam Optimizer [11]. We ran the experiments using TensorFlow [1] and Keras [4].

5 Results and Discussions

Table 1 reports the classification performances of the methods used on the CIFAR-100 sets. Our hybrid loss function performed better than the other losses in most cases and achieved the highest F_1-score in all cases.

Table 2. Performances of Transformer on different loss functions. The high F_1-score and AUC demonstrated that the loss function is suited for the sentiment classification on imbalanced data.

Imb. level (%)	Metrics	Method			
		MFE	MSFE	FL	Hybrid
20	F_1-score	65.56 ± 0.01	67.14 ± 0.01	67.19 ± 0.02	**67.20 ± 0.05**
	AUC	91.67 ± 0.08	91.57 ± 0.09	91.68 ± 0.12	**92.16 ± 0.09**
10	F_1-score	52.57 ± 0.01	54.50 ± 0.01	53.77 ± 0.01	**54.83 ± 0.01**
	AUC	91.04 ± 0.03	90.80 ± 0.04	91.10 ± 0.02	**91.32 ± 0.06**
5	F_1-score	38.07 ± 0.01	40.48 ± 0.01	39.93 ± 0.01	**42.80 ± 0.02**
	AUC	88.93 ± 0.06	88.95 ± 0.02	88.71 ± 0.01	**89.52 ± 0.04**

We report the classification performances of Transformer trained using different loss functions in Table 2. The hybrid loss achieved the highest F_1-score and AUC at all imbalance levels.

6 Conclusion

We studied two loss functions, mean false error and focal loss for training deep neural networks on unbalanced data. As each of the two losses has advantages that can eliminate drawbacks of the other, we showed that hybridizing the two losses in a hybrid loss function that imitates the calculation procedures of mean false error's total loss to focal loss. Tests on this hybrid loss, on image and text classifications, at various imbalance levels, showed that the networks trained with it were superior to mean false error, mean squared false error and focal loss on the F_1-score, but worse in a few cases on the AUC.

This work focused on improving DNN performance for binary classification: future work will evaluate it on multi-class classification.

References

1. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283 (2016)
2. Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S.: An improved algorithm for neural network classification of imbalanced training sets. IEEE Trans. Neural Netw. **4**(6), 962–969 (1993)
3. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw. **106**, 249–259 (2018)
4. Chollet, F., et al.: Keras (2015). https://keras.io
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

6. Hensman, P., Masko, D.: The impact of imbalanced training data for convolutional neural networks. Degree Project in Computer Science, KTH Royal Institute of Technology (2015)
7. Huang, C., Li, Y., Chen, C.L., Tang, X.: Deep imbalanced learning for face recognition and attribute prediction. IEEE Trans. Pattern Anal. Mach. Intell. (2019)
8. Janowczyk, A., Madabhushi, A.: Deep learning for digital pathology image analysis: a comprehensive tutorial with selected use cases. J. Pathol. Inform. **7** (2016)
9. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. J. Big Data **6**(1), 1–54 (2019). https://doi.org/10.1186/s40537-019-0192-5
10. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans. Neural Netw. Learn. Syst. **29**(8), 3573–3587 (2017)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
13. Kudisthalert, W., Pasupa, K., Tongsima, S.: Counting and classification of malarial parasite from giemsa-stained thin film images. IEEE Access **8**, 78663–78682 (2020)
14. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
15. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2537–2546 (2019)
16. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA, June 2011. http://www.aclweb.org/anthology/P11-1015
17. Pasupa, K., Kudisthalert, W.: Virtual screening by a new clustering-based weighted similarity extreme learning machine approach. PLoS ONE **13**(4), e0195478 (2018)
18. Pasupa, K., Vatathanavaro, S., Tungjitnob, S.: Convolutional neural networks based focal loss for class imbalance problem: a case study of canine red blood cells morphology classification. arXiv preprint arXiv:2001.03329 (2020)
19. Sammut, C., Webb, G.I.: Encyclopedia of Machine Learning. Springer Science & Business Media, Berlin (2011)
20. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
21. Wang, H., Cui, Z., Chen, Y., Avidan, M., Abdallah, A.B., Kronzer, A.: Predicting hospital readmission via cost-sensitive deep learning. IEEE/ACM Trans. Comput. Biol. Bioinform. **15**(6), 1968–1978 (2018)
22. Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., Kennedy, P.J.: Training deep neural networks on imbalanced data sets. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 4368–4374. IEEE (2016)
23. Zhu, X., Jing, X.Y., Zhang, F., Zhang, X., You, X., Cui, X.: Distance learning by mining hard and easy negative samples for person re-identification. Pattern Recogn. **95**, 211–222 (2019)