A New Loss Function for Temperature Scaling To Have Better Calibrated Deep Networks

Azadeh Sadat Mozafari[1], Hugo Siqueira Gomes[1], Steeven Janny[2], and Christian Gagné[3]

[1]*{azadeh-sadat.mozafari.1, hugo.siqueira-gomes.1}@ulaval.ca*
[2] *steeven.janny@ens-paris-saclay.fr*
[3]*christian.gagne@gel.ulaval.ca*

Abstract

However Deep neural networks recently have achieved impressive results for different tasks, they suffer from poor uncertainty prediction. Temperature Scaling (TS) is an efficient post-processing method for calibrating DNNs toward to have more accurate uncertainty prediction. TS relies on a single parameter T which softens the logit layer of a DNN and the optimal value of it is found by minimizing on Negative Log Likelihood (NLL) loss function. In this paper, we discuss about weakness of NLL loss function, especially for DNNs with high accuracy and propose a new loss function called *Attended-NLL* which can improve TS calibration ability significantly.

1 Introduction

Deep Neural Networks (DNNs) are able to achieve remarkable performances for different applications, while they suffer from a poor calibration over their uncertainty estimation [15]. Guo et al. [7] investigated this phenomenon and showed that overfitting to Negative Log Likelihood (NLL) loss function during the training phase is the key point to obtain higher accuracy rate during the testing phase. However, the cost of being overfitted to NLL is to loose the calibration of the softmax output layer. Confidence or calibration in deep networks refers to the ability of accurate estimation of true underlying conditional distribution of training data. In real-world applications, calibration is essential for systems security and stability. For instance, in autonomous driving [2] or medical diagnosing systems [11, 4], transferring the control to the human supervisor mostly depends on the confidence over DNNs outputs. Calibrated output also has been shown is useful for rejecting adversaries [3] and out-distributed samples [16, 9] which leads to more secure deep networks.

Deep models calibration methods are studied in recent literature and can be categorized under two main branches: 1-calibration during the training phase, and 2-post-processing based calibration methods. Calibration during training involves an optimization over models that is considering distribution estimation accuracy along with classification accuracy. Well-known training phase calibration methods include approximated Bayesian formalism [5, 18, 19] and variational Bayesian methods [17, 1]. In practice, the quality of predictive uncertainty in Bayesian-based methods relies heavily on the accuracy of sampling approximation and correctly estimated prior distribution [15]. There are also non-Bayesian approaches which approximate the output probability using the ensemble of DNNs [15] or minimizing the calibration error by optimizing a new regularization term during the training phase [14]. However, similarly to Bayesian approaches, these approaches are also involves a significant computational burden, time- and memory-wise.

In contrast, in the second branch of calibration approaches, a pre-trained deep model is calibrated using the post-processing methods. Accordingly, the computational requirements are significantly reduced compared to the first branch. Guo et al. [7] have summarized and compare several famous

NIPS 2018 Workshop on Security in Machine Learning

post-processing calibration methods. They show Temperature Scaling (TS) is the most effective approach that can achieve higher calibration rate with very low computational complexity (optimizing only one parameter T) while preserving the model accuracy. TS algorithm divides the output of logit layer by parameter T to make it softer and more calibrated. The best value of T is found by minimizing the NLL loss function on small set of validation samples respecting to T. NLL is a scoring function [6] which indicates the dissimilarity between the softmax output and the true conditional distribution of data. Optimizing on NLL will allow to find the T at which the logit layer to return high confidence for correctly classified samples and low confidence for wrongly classified samples. But as TS has only one degree of freedom through parameter T, minimizing the NLL, especially for the highly accurate networks, gets dominated by optimizing for the correctly classified samples. Therefore, it will return a T value that increases classification confidence with little consideration for the misclassified samples during the calibration.

Paper Contribution: Our contribution is to propose a new loss function for TS which makes a better adjustment of confidence distribution, not dominated by correctly classified samples. This new loss function, called Attended-NLL, offer a more balanced calibration by some emphasis on two groups of samples: 1)correctly classified with low confidence and 2)wrongly classified samples. We also show that Attended-NLL is a scoring function and minimizing that will make the output of softmax layer more similar to the true conditional distribution of samples. We report the results on NLL and ECE [7] scores which are the standard measures of calibration. We show Attended-NLL loss function improves the result of TS comparing to NLL loss function significantly.

In the following, the notation and problem setting is presented in Sec. 2, followed in Sec. 3 by analysis of the TS method and the issues raised by on the optimization of the NLL loss. Attended-NLL and its scoring function properties are presented in Sec. 4. Finally, results of new proposed TS method with Attended-NLL are reported and discussed in Sec.5

2 Problem Setting

We assume to have access to a pre-trained deep network D with detecting ability of K different classes. There is available validation set $V = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ which is derived from X domain with true marginal distribution $Q(x)$ and conditional distribution $Q(y|x)$. For each sample $x_i \in X$ there exists $\{h_i^1, h_i^2, \ldots, h_i^K\}$ of logit layer outputs which is obtained from last layer of D and the true label $y_i \in \{1, 2, \ldots, K\}$. $y_i \neq k$ means y_i can contain any label except k. For the Temperature Scaling problem, we define the k^{th} softmax output as $S_k(x_i, T) = exp(h_i^k/T)/\sum_{j=1}^{K} exp(h_i^j/T)$ where T is the temperature value.

3 Temperature Scaling with NLL

As the true conditional distribution of the training samples is not available, one of the challenges in calibration is to evaluate the quality of uncertainty estimation obtained from the deep model. NLL is one of the scoring functions which means by minimizing NLL, the output of softmax layer becomes more similar to the true conditional distribution $Q(y|x)$ of samples. NLL is defined as $-\sum_{i=1}^{N} \log S_{y_i}(x_i, T = 1)$ for the softmax layer of a DNN model. In TS method, the logit layer of a deep models is divided by T value in order to rescale the softmax outputs for being more calibrated. The best value of T will be defined by minimizing NLL loss function respecting to T conditioned by $T > 0$ on validation set V as defined in Eq. 1:

$$T^* = \arg\min_T (-\sum_{i}^{N} log(S_{y_i}(x_i, T))) \quad s.t : T > 0 \tag{1}$$

Referring to Guo et al. [7], the Eq. 1 can be rewritten as Eq. 2 for the optimal point T^*:

$$\sum_{i=1}^{N} h_i^{y_i} = \sum_{i=1}^{N} \sum_{k=1}^{K} h_i^k S_k(x_i, T^*) \tag{2}$$

Eq.2 shows when a sample is correctly classified, the weight of h_i^k for the $k = y_i$ which is $S_{y_i}(x_i, T^*)$ should be increased to 1. Therefore T^* goes toward 0. When x_i is misclassified, the weight of h_i^k for

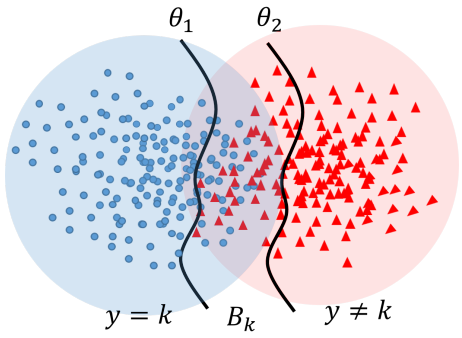

Figure 1: The boundary region B_k is located between tow probability thresholds θ_1 and θ_2.

the $k = y_i$ should be increased to 1 and the weight of wrongly selected label should be decreased. Therefore T^* goes toward ∞ to make $S_k(x_i, T^*)$ close to $1/K$ which is the max possible weight for $h_i^{y_i}$. Because, DNNs are almost highly accurate , the number of correctly classified samples dominates the other samples and has higher impact on the final value of T. If we can increase the impact of misclassified samples, it may help to find more accurate T value. On the other hand, if a sample is correctly classified, NLL tries to give as high as possible confidence to it. However the confidence of correctly classified samples are not always near to 1. If a sample is in the boarder of two classes, it is supposed to have lower conditional probability than a sample which is far from the boundary of the classifier. Based on these two issues, we propose a new loss function which we call it Attended-NLL and it pays more attention on calibration of the misclassified and correctly classified samples which are located in the boundary of the classifier to finetune the T value more accurately.

4 Temperature Scaling with Attended-NLL

The idea of Attended-NLL is to design a scoring function which estimates $Q(x|y = k)$ instead of $Q(y|x)$ which is happened in the case of NLL. This change in distribution estimation gives us the control on samples located in the boundary of the classifier. For data distribution X, conditional distribution $Q(x|y = k)$ can be written as Eq.3 :

$$Q(x|y = k) = \frac{Q(y = k|x)Q(y \neq k)}{Q(y \neq k|x)Q(y = k)} Q(x|y \neq k) \tag{3}$$

This equation demonstrates that the likelihood of sample x given the class k can be estimated from the likelihood of sample x given non-class k. To estimate $Q(x|y = k)$ empirically by Eq. 3, we need enough drawn samples from distribution $Q(x|y \neq k)$ which have a good coverage of X domain especially for the region that $y = k$. As mentioned in [12] this coverage is more probable in the decision boundary of the classifier (boarder of a class). We assume the samples $(x_i, y_i = k)$ which have $S_k(x_i, 1)$ probability smaller than threshold θ_1 and the samples $(x_i, y_i \neq k)$ which have $S_k(x_i, 1)$ bigger than threshold θ_2 are fallen in the decision boundary of the classifier for class k. We call this samples set as B_k. The region B_k is shown in Fig.1. We can rewrite the Eq.3 for the sample of the boundary as:

$$Q(x_i|y_i = k) \approx \frac{S_k(x_i, T)}{1 - S_k(x_i, T)} \alpha_k Q(x_i|y_i \neq k) \quad x_i \in B_k \tag{4}$$

We replaced the $Q(y_i \neq k)/Q(y_i = k)$ with the constant α_k which can be calculated empirically from the samples and approximate $Q(y_i = k|x_i)$ by $S_k(x_i, T)$ and $Q(y_i \neq k|x_i)$ by $1 - S_k(x_i, T)$ functions. We define Attended-NLL loss function as Eq.5 in the boundary region of each class in order to estimate $Q(x|y = k)$:

$$L_{ANLL} = -\sum_{k=1}^{K} \sum_{i=1}^{|B_k|} \log(\frac{S_k(x_i, T)}{1 - S_k(x_i, T)} \alpha_k) \quad x_i \in B_k \tag{5}$$

According to Gibbs inequality, we have $-\mathbb{E}_{Q(x|y)} log Q(x|y) \leq -\mathbb{E}_{Q(x|y)} \log P(x|y)$ for arbitrary distribution $P(x|y)$. Minimum value of $-\mathbb{E}_{Q(x|y)} \log P(x|y)$ is obtained where $P(x|y) = Q(x|y)$.

Table 1: Results of TS calibration with Attended-NLL and NLL loss functions (TS approach keeps the accuracy of the original model unchanged).

			Uncalibrated		**TS NLL**		**TS Attended-NLL**	
Model	**Dataset**	**Accuracy**	ECE%	NLL	ECE%	NLL	ECE%	NLL
DensNet40	CIFAR10	92.66%	4.13	0.301	2.97	0.24	**0.86**	**0.23**
DensNet40	CIFAR100	71.67%	8.48	1.09	1.64	**1.00**	**1.60**	**1.00**
DensNet100	CIFAR10	94.66%	2.97	0.23	4.96	0.20	**0.72**	**0.17**
DensNet100	CIFAR100	76.45%	12.00	1.12	17.76	**1.03**	**11.68**	1.11
DensNet100	SVHN	95.87%	1.94	0.17	1.14	0.16	**0.77**	**0.16**
ResNet110	CIFAR10	93.60%	4.31	0.30	4.02	0.23	**1.80**	**0.21**
ResNet110	CIFAR100	70.30%	13.17	1.26	**1.86**	**1.06**	2.44	**1.06**
ResNet110	SVHN	95.53%	3.14	0.22	3.18	0.18	**0.82**	**0.16**
WideResNet110	CIFAR100	74.40%	14.17	1.23	2.73	0.93	**1.97**	**0.92**

Table 2: The selected values for different models and datasets parameters for TS with NLL and with Attended-NLL approaches

DNN Model	Dataset	T_{NLL}	T_{ANLL}	θ_1	θ_2
DensNet40	CIFAR10	2.44	1.863	0.99	0.1
DensNet40	CIFAR100	1.426	1.422	0.86	0.1
DensNet100	CIFAR10	2.855	1.847	0.99	0.01
DensNet100	CIFAR100	2.954	1.02	0.99	0.1
DensNet100	SVHN	1.222	1.379	0.94	0.85
ResNet110	CIFAR10	2.92	1.913	0.995	0.999
ResNet110	CIFAR100	1.826	1.623	0.95	0.9
ResNet110	SVHN	2.278	1.884	0.99	0.93
WideResNet110	CIFAR100	2.206	2.02	0.97	0.92

By considering $Q(x|y=k) \approx Q(x|y \neq k)$ in B_k region, and based on Gibbs inequality, minimizing Attended-NLL respecting to T on samples of B_k set, leads to $S_k(x_i, T^*)/(1 - S_k(x_i, T^*))$ to be similar to $Q(y_i = k|x_i)/(1 - Q(y_i = k|x_i))$. It means $S_k(x, T^*)$ should be similar to $Q(y = k|x)$ on B_k. Therefore, Attended-NLL loss function is a scoring function which can calibrate the softmax output layer of the DNN for the samples of B_k region. There is a concern that Attended-NLL only focuses on B_k samples to calibrate the output and do not consider all the samples of the training domain. This may cause misleading in true conditional distribution estimation. But it should be considered that a pre-trained DNN model is already overfitted to NLL loss function during the training phase and applying TS with Attended-NLL is just to finetune the output distribution of DNN model for the B_k samples and not estimating the output distribution from scratch.

5 Experiments

We apply TS calibration method with two different loss functions (NLL and Attended-NLL) for different DNN models and datasets. The selected models are DenseNet with depth 40 and 100 [10], ResNet with depth 110 [8] and WideResNet with depth 32 [21] which are trained on training set of several standard classification datasets (CIFAR10/100 [13] and SVHN [20]). Later, we select 3000 validation images randomly from validation sets of each dataset and apply TS with NLL and Attended-NLL on them to find the best T values. In TS with Attended-NLL to select the boundary sample set B_k, we select the best values for θ_1 and θ_2 parameters respecting to lowest value for NLL on validation set. The evaluation of calibration quality is reported on testing samples of datasets. Standard measures ECE with 15 bins [7] and NLL scores are considered for comparison in which lower means more calibrated. As shown in Table 1, TS with Attended-NLL loss function achieves better calibration results for both NLL and ECE measures for almost all models and datasets. As we have discussed in section 4, Attended-NLL can improves calibration significantly for the models which have higher accuracy rates. Table 2 gives the information about the T values for two different loss functions and the thresholds which define the boundary region of Attended-NLL approach. It shows the small change in T value can have significant change in calibration rate, therefore finetuning the T parameter is important regarding to have more accurate calibration.

References

[1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[3] John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

[4] Cynthia S Crowson, Elizabeth J Atkinson, and Terry M Therneau. Assessing calibration of prognostic risk scores. *Statistical methods in medical research*, 25(4):1692–1706, 2016.

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[6] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[11] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19 (2):263–274, 2011.

[12] Long Jin, Justin Lazarow, and Zhuowen Tu. Introspective classification with convolutional nets. In *Advances in Neural Information Processing Systems*, pages 823–833, 2017.

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

[14] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2810–2819, 2018.

[15] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[16] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[17] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016.

[18] David JC MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

[19] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[20] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[21] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.