1 Introduction

In high-stakes tasks, a reliable model should not only make predictions but give guarantees (e.g. indicate when they are likely to be incorrect). However, traditional deep learning models are ill-equipped to evaluate the uncertainty of their decisions. *Model calibration* solve this problem by approximating the predicting probability to the true correctness likelihood. This project focuses on: **how model calibrates when employed with different loss function?**

2 Preliminaries

2.1 Problem Setup

We consider a standard supervised multi-class classification problem in which we observe examples $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y} = \{1, 2, \cdots, K\}$ is the label space with K classes, and aim to predict y from $\boldsymbol{x}$. Here we assume $\boldsymbol{f}^\theta$ as a composition of a non-probabilistic K-way classifier $\boldsymbol{g}^\theta$ and a softmax function σ, i.e. $\boldsymbol{f}^\theta = \boldsymbol{g}^\theta \circ \sigma$. For a query instance x, $\boldsymbol{f}^\theta$ gives its probability of assigning it to label i as $\exp(g_i^\theta(\boldsymbol{x}))/\sum_{k=1}^K \exp(g_i^\theta(\boldsymbol{x}))$, where $\boldsymbol{g}_i^\theta(\boldsymbol{x})$ denotes the i-th element of the logit vector produced by $\boldsymbol{g}^\theta$. Then, $\hat{y} := \arg\max_i \boldsymbol{f}_i^\theta(\boldsymbol{x})$ can be returned as the predicted label and $\hat{p} := \max_i \boldsymbol{f}_i^\theta(\boldsymbol{x})$ can be treated as the associated confidence score.

2.2 Metric

One notion of miscalibration is the difference in expectation between confidence and accuracy, i.e.

$$E[|P\{y|\boldsymbol{x}\} - \hat{p}|] \tag{1}$$

Expected Calibration Error approximates (1) by partitioning predictions into M equally-spaced bins (similar to the reliability diagrams) and taking a weighted average of the bins' accuracy/confidence difference. More precisely,

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)| \tag{2}$$

where n is the number of samples. We will use ECE as the primary empirical metric to measure calibration in our study.

2.3 Loss Function

L_p Norm in the Function Space is one of the explicit regularization methods for calibration investigated by the recent work. For a real number $p \geq 1$, the L_p Norm of a vector $\boldsymbol{z}$ with dimension n can be expressed as: $||\boldsymbol{z}||_p = (\sum_i^n |\boldsymbol{z}_i|^p)^{1/p}$. By adding L_p Norm of logits $\boldsymbol{g}^\theta$ with a weighting coefficient α into final objective function.

$$\mathcal{L}_{L_p}(y, \boldsymbol{f}^\theta) = \mathcal{L}_{ce}(y, \boldsymbol{f}^\theta) + \alpha||\boldsymbol{g}^\theta||_p$$

The function complexity of neural networks can be directly penalized during training.
Entropy Regularization directly penalizes predictive distributions that have low entropy, prevents these peaked distributions, and ensures better model generalization.

$$\mathcal{L}_{ER}(y, \boldsymbol{f}^\theta) = \mathcal{L}_{ce}(y, \boldsymbol{f}^\theta) - \alpha H(\boldsymbol{f}^\theta)$$

Focal Loss is originally proposed to address the class imbalance problem in object detection. By reshaping the standard CE loss through weighting loss components of all samples according to how well the model fits them, focal loss focuses on fitting hard samples and prevents the easy samples from overwhelming the training procedure. Formally, for classification tasks where the target distribution is one-hot encoding, it is defined as

$$\mathcal{L}_f = (1 - \boldsymbol{f}_y^\theta)^\gamma \log \boldsymbol{f}_y^\theta$$

where γ is a predefined coefficient.

3 Experiments

3.1 Experiment Setup

We conduct a comparison study of the above regularization methods on Cifar-10. We train ResNet18, ResNet50 and VGG16 with the standard CE loss and the above regularized losses respectively. For Norm regularization, we use L1 Norm, which has been shown effective for calibration despite its simple form.

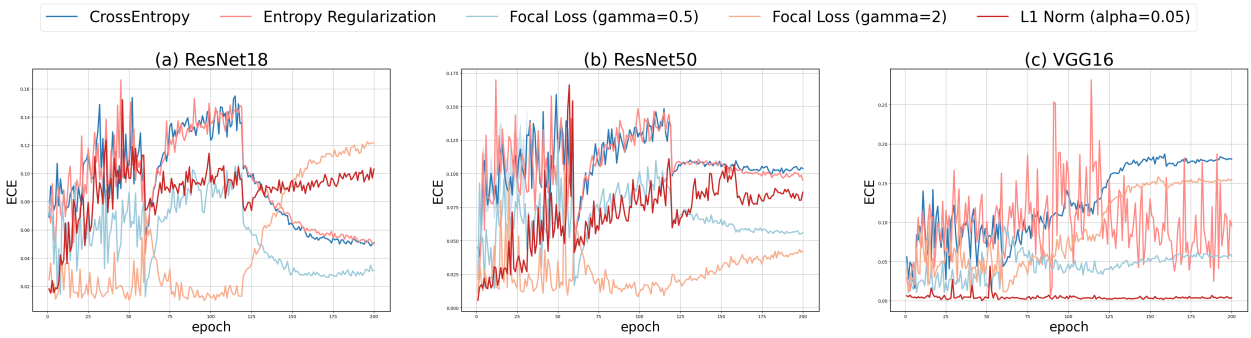

Figure 1: comparison results of ECE.

3.2 Results

Cross Entropy(NLL) We observe that models trained with cross entropy loss are prone to over-confidence, which is linked to overfitting on the negative log-likelihood (NLL). Concretely, along with the standard NLL-based model training, when classification error is minimized, keeping training will further push the model to minimize NLL on the training data, i.e., push the predicted softmax probability distribution as close as possible to the ground-truth distribution (which is usually one-hot). Model overfitting starts by exhibiting increased test NLL, and then the model becomes overconfident.

Entropy Regularization The results show that entropy regularization is not effective to mitigate over-confidence. Moreover, on VGG16, we notice a strange phenomenon that the ECE value fluctuates a lot (unlike other loss functions).

L1 Norm We can see L1 Norm alleviates the over-confidence (except on ResNet18). This can be explained by the following theorem:

Theorem 1. *For any given logits* $(f_1, \cdots, f_n)$*, where* $f_1 > f_2 > \cdots > f_n$*, and constant* $t > 1$*, we have*

$$\frac{e^{tf_1}}{\sum_{j=1}^{n} e^{tf_j}} \geq \frac{e^{f_1}}{\sum_{j=1}^{n} e^{f_j}}$$

Proof. Let $t = 1 + s$. Then, we have

$$\frac{e^{tf_1}}{\sum_{i=1}^{n} e^{tf_i}} = \frac{e^{(1+s)f_1}}{\sum_{i=1}^{n} e^{(1+s)f_i}} = \frac{e^{f_1}}{\sum_{i=1}^{n} e^{f_i} e^{s(f_i - f_1)}} > \frac{e^{f_1}}{\sum_{i=1}^{n} e^{f_i}}$$

□

The L_1 Norm penalty helps constrain the magnitude of logits $\boldsymbol{g}^{\theta}$, ensuring that, after applying the softmax function, excessive confidence $\boldsymbol{f}^{\theta}$ are avoided.

Focal Loss The figure demonstrates that models trained with focal loss often lead to lower ECE. This is because focal loss is upper bound by the regularised KL-divergence, where the regulariser is the negative entropy of the predicted probability.

Theorem 2.

$$\mathcal{L}_f(y, \boldsymbol{f}^\theta) \geq \mathcal{L}_{ce}(y, \boldsymbol{f}^\theta) - \gamma H(\boldsymbol{f}^\theta)$$

Proof.

$$\begin{aligned}
\mathcal{L}_f(y, \boldsymbol{f}^\theta) &= -\sum_{y=1}^{K}(1 - \boldsymbol{f}_y^\theta)^\gamma q_y \log \boldsymbol{f}^\theta && (3)\\
&\geq -\sum_{y=1}^{K}(1 - \gamma \boldsymbol{f}_y^\theta) q_y \log \boldsymbol{f}^\theta && (4)\\
&= -\sum_{y=1}^{K} q_y \log \boldsymbol{f}^\theta - \left|\sum_{y=1}^{K} \gamma \boldsymbol{f}_y^\theta q_y \log \boldsymbol{f}^\theta\right| && (5)\\
&\geq -\sum_{y=1}^{K} q_y \log \boldsymbol{f}^\theta - \gamma \max_j q_j \sum_{y=1}^{K} |q_y \log \boldsymbol{f}^\theta| && (6)\\
&\geq -\sum_{y=1}^{K} q_y \log \boldsymbol{f}^\theta - \gamma \sum_{y=1}^{K} |q_y \log \boldsymbol{f}^\theta| && (7)\\
&= \mathcal{L}_{ce}(y, \boldsymbol{f}^\theta) - \gamma H(\boldsymbol{f}^\theta) && (8)
\end{aligned}$$

q_y denotes the one-hot probability. □

Therefore, trying to minimise focal loss minimises the cross-entropy loss, whilst simultaneously increasing the entropy of the predicted distribution $\boldsymbol{f}^\theta$.