1 Abstract

Uncertainty quantification in deep learning is essential for assessing the reliability of model predictions. Despite their success in various tasks, neural networks often fail to provide meaningful measures of uncertainty, limiting their trustworthiness. We explore model-related uncertainty in neural networks, focusing on calibration techniques that align predicted probabilities with true frequencies, thus improving the interpretability and reliability of predictions.

Experiments were conducted using a ResNet-18 model trained on the CIFAR-10 dataset. Various loss functions and additional terms were employed to assess their impact on model calibration and expected calibration error (ECE). Results indicate that specific modifications can enhance calibration without significantly compromising accuracy, evidenced by reduced ECE values during validation.

This research is expected to contribute to the theoretical understanding of uncertainty in deep learning and proposes directions for improving uncertainty quantification, particularly in areas where accurate risk assessment is critical.

2 Introduction

Deep learning has transformed fields such as computer vision, natural language processing, and autonomous systems, but the reliability of neural network predictions remains a concern, especially in high-stakes applications like healthcare and finance. A key challenge is the quantification of uncertainty, which is essential for informed decision-making and effective risk management.

Calibration, the alignment between predicted probabilities and actual outcomes, is crucial for reliable predictions. A well-calibrated model produces probabilities that reflect true likelihoods. For instance, if a model predicts 0.8 for a class, it should be correct approximately 80% of the time. Unfortunately, state-of-the-art neural networks are often poorly calibrated, leading to overconfident predictions due to factors like model architecture, training data distribution, and optimization techniques [5]. Techniques such as Platt scaling and isotonic regression address this by adjusting probabilities post-training [1].

Uncertainty in model predictions can be categorized into *aleatoric* (stemming from data noise) and *epistemic* (resulting from model limitations and limited data) [2]. Distinguishing between these types informs whether uncertainty can be reduced through additional data or

improved models.

Recent work [3] proposes a theoretical framework decomposing *pointwise risk* into *Bayes risk* and *excess risk*, using Bregman divergences to analyze epistemic uncertainty. This helps interpret model uncertainties and guides adjustments to training processes and architectures. Rigorous evaluation metrics like Expected Calibration Error (ECE) measure deviations between predicted probabilities and actual outcomes, aiding the study of calibration and its effects on prediction reliability [1].

This thesis investigates model-related uncertainty in neural networks, focusing on calibration techniques and their impact on ECE in a ResNet-18 model trained on CIFAR-10. By exploring the influence of loss functions and additional terms, this work aims to enhance calibration and advance theoretical understanding of uncertainty in DL models.

3 Problem statement

We consider the problem of supervised multiclass classification with NNs.

The data features $X \in \mathcal{X}$ and labels $Y \in \mathcal{Y} = \{1, \dots, K\}$ are random variables with ground truth distribution:

$$\pi(X, Y) = \pi(Y|X) \cdot \pi(X)$$

Let $h : \mathcal{X} \to \mathcal{Y} \cdot [0, 1]$ be a model of classification, which returns a predicted label and its probability: $h(X) = (\widehat{Y}, \widehat{P})$. We want the model to be calibrated.

Perfectly calibrated classification model — a model $h : \mathcal{X} \to \mathcal{Y} \times [0, 1]$, such that

$$\mathbb{P}_{(X,Y)\sim\pi}(\widehat{Y} = Y|\widehat{P} = p) = p \ \forall p \in [0, 1] \tag{1}$$

Let us denote: $h_(x) = (f(x), q(x))$. Then we have:

$$\mathbb{P}(f(X) = Y|q(X) = p) = \mathbb{E}_{X,Y}\left[I\left\{f(X) = Y\right\}|q(X) = p\right] =$$

$$= \mathbb{E}_X\left[\mathbb{E}_Y\left[I\left\{f(X) = Y\right\}|X\right]|q(X) = p\right] = \mathbb{E}_{X,Y}\left[P(Y = f(X))|q(X) = p\right]$$

So, it is the averaged accuracy over $x \in q^{-1}(p)$

3.1 Risk decomposition

There are 2 types of uncertainty. The first, *aleatoric*, comes from the inherent ambiguity in label y distribution, given the x. So, it comes from $\pi(y|x)$. The second, *epistemic*, is

believed to came from the "lack of knowledge". Hence, it is defined by the model capacity and amount of training data. [3]

Let us denote $D_{tr} = \{(X_i, Y_i)\}_{i=1}^N$ - train dataset, where pairs $X_i \in \mathbb{R}^d, Y_i \in \mathcal{Y} = \{1, \dots, K\}$ are i.i.d from a joint distribution $P_{tr}(X, Y)$. Also, let us define $\eta(x) = \mathbb{P}(Y|X = x)$ to be real distribution, $\widehat{\eta}(x)_\theta = \widehat{\eta}(x) = \mathbb{P}(Y|X = x, \theta)$ — the distribution, approximated by our model. Let $l : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ be a loss function, measuring how well K-categorial distribution, given by our model $\widehat{\eta}(x)_\theta$ far from x's real label y.

But actually, due to *aleathoric* ancertainty we have not the one y, but a distribution over y-s: $\mathbb{P}(y|X = x) = \eta(x)$. So, it is sensible to introduce a pointwise risk, that will average all losses by y-s [3]:

$$R(\widehat{\eta}|x) = \int l(\widehat{\eta}_\theta(x), y) d\mathbb{P}(y|X = x)$$

Then, author decompose the total risk into *Bayes* and *Excess*:

$$R(\widehat{\eta}_\theta|x) = R_{\text{Bayes}}(x) + R(\widehat{\eta}_\theta|x) - R_{\text{Bayes}}(x)$$

Where pointwise *Bayes* is defined as:

$$R_{\text{Bayes}}(x) = \int l(\eta(x), y) dP(y|X = x)$$

So, it is the loss of data generate process, that can not be reduced, so, we may call him aleathoric, while excess term we may call epistemic [3].

Then authors of the [3] introduce the concept of *strictly proper* scoring rules. Let $P \in \mathcal{P}_K$ - be a K-categorical distribution, $l(P, y) : \mathcal{P}_K \times Y \to \mathbb{R}$ be a loss function. Averaging it for the distribution under y-s Q we will have $l(P, Q) = \int l(P, y) dQ(y)$. So, the scoring rule l called *scrictly proper*, iff $\forall P, Q \in \mathcal{P}_K : l(P, Q) \geq l(Q, Q)$, and equality iff $Q = P$a.s. [3]. After that, authors of [3] proove, that under some assumptions, any strictly proper scoring rule can be represented as:

$$l(\eta, i) = \langle G'(\eta), \eta \rangle - G'_i(\eta) - G(\eta)$$

for some strictly convex scalar function $G : \mathcal{P}_K \to \mathbb{R}$.

Let us also denote a full vector of errors:

$$l(\eta) = \begin{pmatrix} l(\eta, 1) \\ l(\eta, 2) \\ \dots \\ l(\eta, K) \end{pmatrix} = (\langle G'(\eta), \eta \rangle - G(\eta)) \cdot \mathbb{1} - G'(\eta)$$

.

It is obvious, that

$$R_{Tot}(\widehat{\eta}) = \langle l(\widehat{\eta}), \eta \rangle$$

$$R_{Tot}(\widehat{\eta}|x) = \langle l(\widehat{\eta}(x)), \eta(x) \rangle = \int l(\widehat{\eta}(x), y) dP(y|X = x)$$

3.2 Risk decomposition

So, we have:

$$l(\eta, y) = \langle G'(\eta), \eta \rangle - G'_y(\eta) - G(\eta)$$

.

Let us call this term as total OHE risk ($y \in \{1, \ldots, K\}$):

$$R_{Tot}^{OHE} = l(\widehat{\eta}, y) = \langle l(\widehat{\eta}), ohe(y) \rangle = \langle l(\widehat{\eta}), ohe(y) - \eta \rangle + \langle l(\widehat{\eta}), \eta \rangle = \langle l(\widehat{\eta}), ohe(y) - \eta \rangle + R_{Tot}$$

$$R_{Tot} = \langle l(\widehat{\eta}), \eta \rangle$$

Ideally, we want to optimize only R_{Tot}. But, since we do not know η, we usually optimize R_{Tot}^{OHE} (so, we usually approximate the term $\eta(x)$ by $ohe(y)$.

So, let us decompose a middle term:

$$\langle l(\widehat{\eta}), ohe(y) - \eta \rangle = \langle (\langle G'(\widehat{\eta}), \widehat{\eta} \rangle - G(\widehat{\eta})) \cdot \mathbb{1} - G'(\widehat{\eta}), ohe(y) - \eta \rangle =$$

$$= (\langle G'(\widehat{\eta}), \widehat{\eta} \rangle - G(\widehat{\eta})) \cdot \langle \mathbb{1}, ohe(y) - \eta \rangle - \langle G'(\widehat{\eta}), ohe(y) - \eta \rangle = -\langle G'(\widehat{\eta}), ohe(y) - \eta \rangle$$

(the last is because sum of components of the vector $ohe(y) - \eta$ equals 0: they are difference between two distributions).

Finally, we got:

$$R_{Tot} = R_{Tot}^{OHE} + \langle G'(\widehat{\eta}), ohe(y) - \eta \rangle$$

Let us try to approximate the term $ohe(y) - \eta$ by discovering its properties (and by this way approximate the total risk).

3.3 Total risk approximation

Let us denote $s = ohe(y) - \eta$. Here we will study its properties.

1. $\langle \mathbb{1}, s \rangle = 0$.
2. $\exists! i : s_i \geq 0 \wedge \forall j \neq i : s_j \leq 0$. $s_i \in [-1, 0], s_j \in [0, 1]$.
3. There is no gradients of s with respect to $\widehat{\eta}$.
4. $\mathbb{E}_{y \sim \eta(y|x)} s = 0$, since $\mathbb{E}_{y \sim \eta(y|x)} ohe(y) = \eta$.

3.3.1 Uniform approximation

The first simple idea is to use uniform approximation. So, all components of s are equal, except one:

$$s_k = \begin{cases} \varepsilon(1 - \frac{1}{K}), & if\ y = k \\ -\frac{\varepsilon}{K}, & else \end{cases}$$

Properties (1-3) are satisfied, and property (4) is satisfied for uniform η: $\eta(x) = \frac{1}{K}$:

$$\mathbb{E}_{y \sim \eta} s_k(y) = \varepsilon(1 - \frac{1}{K}) \cdot \eta_k - \varepsilon \frac{1}{K}(1 - \eta_k) = 0 \implies \eta_k = \frac{1}{K}$$

3.3.2 Prior approximation

$$s_k = \begin{cases} \varepsilon(1 - \frac{N_k}{N}), & if\ y = k \\ -\varepsilon \frac{N_k}{N}, & else \end{cases}$$

The property (4) holds for prior η: $\eta = \frac{N_k}{N}$:

$$\mathbb{E}_{y \sim \eta} s_k(y) = \varepsilon(1 - \frac{N_k}{N}) \cdot \eta_k - \varepsilon \frac{N_k}{N}(1 - \eta_k) = 0 \implies \eta_k = \frac{N_k}{N}$$

3.3.3 Predicted approximation

Here SG means *stopgrad*.

$$s_k = \begin{cases} \varepsilon(1 - SG(\widehat{\eta}_k)), & if\ y = k \\ -\varepsilon SG(\widehat{\eta}_k), & else \end{cases}$$

Here property (4) satisfied iff $\eta = \widehat{\eta}$.

So, substituting different s we will get different approximations of total risk and, possibly, better calibrated model than with usage only OHE risk.

Table 1: Comparison of Metrics

	Log Score	**Brier Score**
$G(\eta)$	$\sum_{k=1}^{K} \eta_k \log \eta_k$	$-\sum_{k=1}^{K} \eta_k(1-\eta_k)$
$G'(\widehat{\eta})_k$	$1 + \log \widehat{\eta}_k$	$2\widehat{\eta}_k - 1$

4 Experiments

4.1 Table with metrics

To calculate loss and addition term, let's derive formulas for the G function and G' for 2 most common losses in classification task:

4.2 Experiment setup

Experiments were done with losses: log score and brier score and with s-approximations of StopGrad, Prior and Uniform on the dataset CIFAR-10 on the ResNet-18 model throughout 50 epochs.

We measured values of:

1. *Accuracy-top-1*: how frequent the label with highest predicted probability equals the true label.

2. *Expected Calibration Error*: measures, how well discretized probabilities align with true probabilities. Is is the difference between the fraction of predictions in the bin that are correct (accuracy) and the mean of the probabilities in the bin (confidence) [4]. ECE compites weighted average error across bins:

$$ECE = \sum_{b=1}^{B} \frac{n_b}{N} |acc(b) - conf(b)|$$

   Where n_b — number of predictions in bin b, N is the total number of data items, $acc(b)$ and $conf(b)$ are accuracy and confidence respectively.

3. *Loss*. Either CrossEntropyLoss or BrierLoss.

4. *Addition* loss addition, which is defined by the choise of s.

The theoretical hypothesies we want to test:

1. Addition term helps construct better-calibrated model. So, with this term ECE should be less.

2. Accuracy does not degradate significantly when introducing addition term. So, the model remains accurate for task.

3. ECE of the model is a convex function of epoch/accuracy: it first decreases, then increases. This corresponds to the follofing proposition: at the beginning of the training OHE loss is co-directed with calibration, towards the end it pushes the model to be less calibrated and more accurate.

4.3 Experiment results

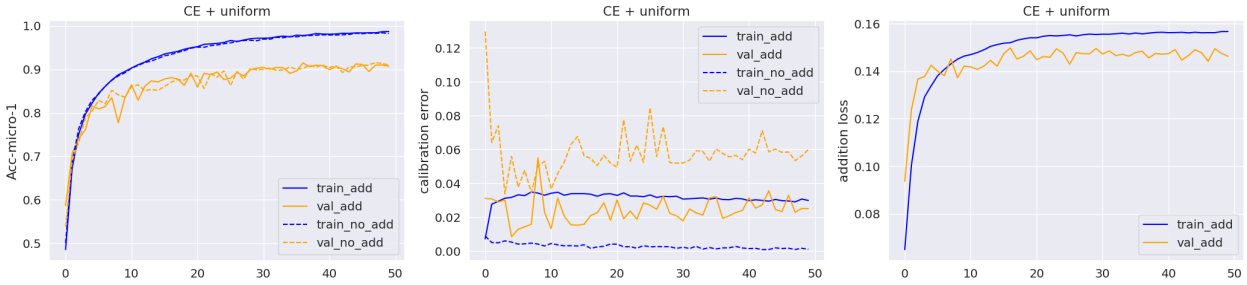

Figure 1: Cross Entropy Loss and Uniform addition

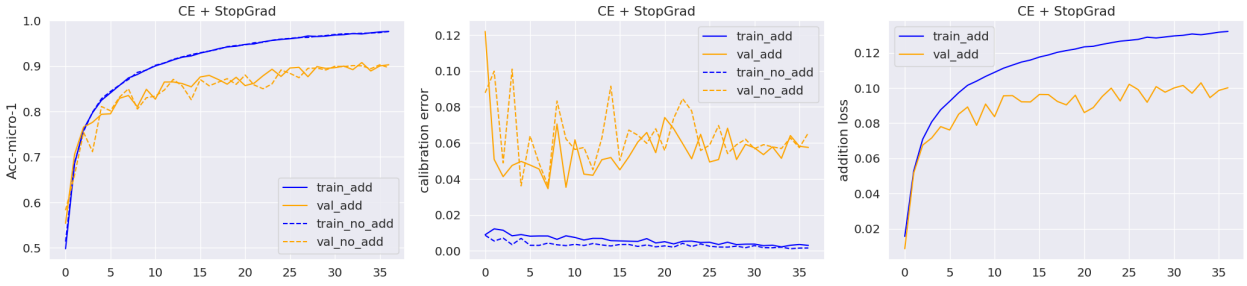

Figure 2: Cross Entropy Loss and StopGrad addition

Figure 3: Brier Loss and Uniform addition

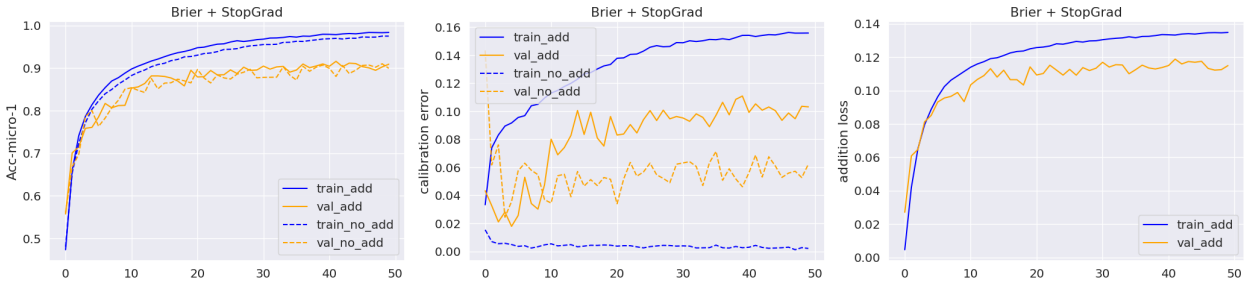

Figure 4: Brier Loss and StopGrad addition

Observations:

1. All experiments are comparable, because exhibit similar behaviour (accuracy plot).
2. No difference in accuracy with and without addition.
3. Addition loss is growing throughout the all learning process and plateaued. It exhibits similar behaviour to accuracy.
4. On 2-nd and 3-rd settings we observe the same behaviour of ECE:
   - it is higher on validation and lower on training, indicating a overtraining in terms of calibration. So, model is more confident on unseen examples in validation dataset than on examples in training dataset.
   - it plummets on validation at the beginning and then steadily increase (with flictuations)
   - it does not changes on train (can not understand)
5. On 1-st and 4-th graph we observe that addition deteriorates calibration in train (1-st) and in train and validation (4-th)

Conclusion: 2-nd hypothesis proved to be correct, 3-rd seems to be true, but more experiments should be conducted. 1-st is under question.

Future experiments: to validate our hypothesies we should conduct very simple experiment, where every distribution can be easily inferred to validate all hypothesies.

4.4 Risk annigilation

Let $D = \{X, Y\}, x_i \in \mathbb{R}^d, y_i \in \{1, \dots, K\}.(x, y) \sim \pi(x) \cdot \eta(y|x)$. So, $\eta : \mathbb{R}^d \to \Delta^K$ is a real probability, we want to learn.

Ideally, we want to optimize:

$$\mathsf{R}_{\text{Tot}} = \mathsf{R}^{\text{Tot}}_{l,D}(\eta, \widehat{\eta}) = \sum_{i=1}^{n} l(\widehat{\eta}(x_i), \eta(x_i))$$

Let us define: $\eta^* = \arg\min_{\widehat{\eta}} \mathsf{R}^{\text{Tot}}_{l,D}(\eta, \widehat{\eta})$. Our goal is to find η^* through gradient optimization, so, we only care about the calculation of gradients: $\frac{\partial \mathsf{R}^{\text{Tot}}_{l,D}(\eta, \widehat{\eta})}{\partial \widehat{\eta}}$.

But as we do not know $\eta(x_i)$, but know that this distribution has high probability in y_i, we make a substitution: $\eta(x_i) \to \text{ohe}(y)$ (into one-host distribution) and obtain the following risk:

$$\mathsf{R}^{\text{OHE}}_{\text{Tot}} = \mathsf{R}^{\text{Tot, OHE}}_{l,D}(\widehat{\eta}) = \sum_{i=1}^{n} l(\widehat{\eta}(x_i), \text{ohe}(y_i))$$

And we have an access only to its ohe gradients: $\frac{\partial \mathsf{R}^{\text{Tot, OHE}}_{l,D}(\widehat{\eta})}{\partial \widehat{\eta})}$,

Let us use the following representation:

$$\begin{aligned}\mathsf{R}^{\text{Tot, OHE}}_{l,D}(\widehat{\eta}) = \sum_{i=1}^{n} l(\widehat{\eta}(x_i), \eta(x_i)) + \\ + \sum_{i=1}^{n} [l(\widehat{\eta}(x_i), \text{ohe}(y_i)) - l(\widehat{\eta}(x_i), \eta(x_i))] = \\ = \mathsf{R}^{\text{Tot}}_{l,D}(\eta, \widehat{\eta}) + \left(\mathsf{R}^{\text{Tot, OHE}}_{l,D}(\widehat{\eta}) - \mathsf{R}^{\text{Tot}}_{l,D}(\eta, \widehat{\eta})\right) \quad (2)\end{aligned}$$

Let us call the first term as *calibration* loss. The goal is to find its gradient w.r.t. $\widehat{\eta}$. We can not do it explicitly, but can use the following trick:

1. Let us use several different loss functions: $l_1, \ldots, l_T$

2. Let us average the following expression w.r.t. these loss functions, so, get:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} \mathsf{R}^{\text{Tot, OHE}}_{l_t,D}(\widehat{\eta}) = \frac{1}{T}\sum_{t=1}^{T} \mathsf{R}^{\text{Tot}}_{l_t,D}(\eta, \widehat{\eta}) + \\ + \sum_{t=1}^{T} \left(\mathsf{R}^{\text{Tot, OHE}}_{l_t,D}(\widehat{\eta}) - \mathsf{R}^{\text{Tot}}_{l_t,D}(\eta, \widehat{\eta})\right) \quad (3)\end{aligned}$$

3. As the first term responsible for calibration, and we use proper scoring rules for l_t, the T gradients of the first term should be co-directed (as they all point out to the η^*). So, intuitively, gradients of first term (or their direction) do not depend on the l_t, since they are all proper scoring rules.

4. In contrast, the second term, a noise term, or approximation error, intuitively much more dependent on the l_t and thus gradients in its summation will be differently oriented and annigilate in summation.

5. As a result, averaging total risk across many proper losses, we will be able to get rid of our approximation error and more precisely calculate gradients of our real risk $\mathsf{R}^{\text{Tot, OHE}}$.

Theorem 1. *Usage of multiple losses, obtained from proper scoring rules, do not annigilate the desired term.*

Proof. This term (after all contractions) is:

$$f(\eta, y, x) = \langle G'(\widehat{\eta}(x)), ohe(y) - \eta(x) \rangle = \langle G'(\widehat{\eta}), ohe(y) - \eta \rangle$$

Its gradient wrt $\widehat{\eta}$:

$$\nabla_{\widehat{\eta}}(f) = G''(\widehat{\eta}) \cdot (ohe(y) - \eta)$$

□

4.5 Experiments

5 Thoughts

But I insist, that we can calculate the 1:

$$\mathbb{P}(f(x) = y | x \in A) = \frac{\mathbb{P}(f(x) = y \wedge x \in A)}{\mathbb{P}(x \in A)};$$

$$\mathbb{P}(f(x) = y \wedge x \in A) = \mathbb{E}_{(x,y)\sim\pi} I\left[x \in A \cap f^{-1}(y)\right] = \mathbb{E}_{y\sim\pi_y(y)}\left[\mathbb{E}\left[I\left[x \in A \cap f^{-1}(y)\right] | Y = y\right]\right] =$$

$$= \mathbb{E}_{y\sim\pi_y(y)} \mathbb{P}_{X|Y}(x \in A \cap f^{-1}(y) | Y = y) = \sum_{y=1}^{K} \mathbb{P}(Y = y) \cdot \mathbb{P}(X \in A \cap f^{-1}(y) | Y = y)$$

Finally, we have:

$$\mathbb{P}(f(x) = y | x \in A) = \frac{\sum\limits_{y=1}^{K} \mathbb{P}(Y = y) \cdot \mathbb{P}(X \in A \cap f^{-1}(y) | Y = y)}{\mathbb{P}(X \in A)}$$

References

[1] Nikita Durasov et al. “Zigzag: Universal sampling-free uncertainty estimation through two-step inference”. In: *arXiv preprint arXiv:2211.11435* (2022).

[2] Chuan Guo et al. “On calibration of modern neural networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 1321–1330.

[3] Nikita Kotelevskii and Maxim Panov. “Predictive Uncertainty Quantification via Risk Decompositions for Strictly Proper Scoring Rules”. In: *arXiv preprint arXiv:2402.10727* (2024).

[4] Jeremy Nixon et al. “Measuring Calibration in Deep Learning.” In: *CVPR workshops*. Vol. 2. 7. 2019.

[5] Cheng Wang. “Calibration in deep learning: A survey of the state-of-the-art”. In: *arXiv preprint arXiv:2308.01222* (2023).