uc3m | Universidad Carlos III de Madrid

Master in Machine Learning for Health,

Deep Learning
2025-2026

Project I: Understanding Calibration in Convolutional Neural Networks (CNNs)

Santiago Prieto Núñez

Alex Sánchez Zurita

Jorge Barcenilla González

20/10/2025

INTRODUCTION

The wide-extended neural networks models have a necessity of confidence calibration. This is due to the core problem that this project and the paper *On Calibration of Modern Neural Networks* by *Chuan Guo Geoff Pleiss Yu Sun Kilian Q. Weinberger* addresses.

A model that is 99% confident should, in fact, be correct 99% of the time. When this is not the case, the model is considered mis-calibrated, which can lead to silent failures, misplaced user trust, and bad outcomes in decision-making pipelines.

The purpose of this report is to present a practical case study demonstrating the implementation and effectiveness of **temperature scaling**, a simple post-processing technique designed to correct this **miscalibration**. This report details an experiment where a LeNet5 model was trained for a binary classification task (distinguishing birds from cats) and subsequently calibrated, measuring the impact on its confidence reliability.

This experimental study evaluates three distinct models to assess calibration performance under different training conditions. The first model employs a baseline approach without regularization techniques, serving as a control. The second incorporates regularization methods to improve generalization.

THEORY & TEMPERATURE SCALING

Temperature scaling is a post-processing calibration technique that addresses model miscalibration by **introducing a scalar** parameter $T > 0$ (called temperature) **into the softmax** function. This method was proposed by Guo et al. as a single-parameter variant of Platt scaling and has proven remarkably effective despite its simplicity.

Mathematical Formulation

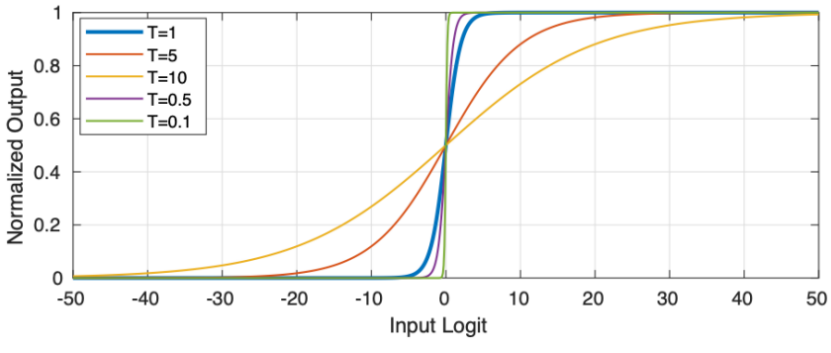
For a multiclass classification problem with K classes, let $\mathbf{z}_i \in \mathbb{R}^K$ denote the logit vector (pre-softmax output) produced by the neural network for input x_i. The standard softmax function produces class probabilities as:

$$p_k = \sigma_{SM}(\mathrm{z_i})^{(k)} = \frac{\exp\left(z_i^{(k)}\right)}{\sum_{j=1}^{K} exp(z_i)}$$

Temperature scaling modifies this by dividing the logits by temperature T:

$$\hat{q}_i = \max_k \ \sigma_{SM}\left(\frac{z_i}{\boldsymbol{T}}\right)^{(k)}$$

where $\hat{q}_i$ represents the calibrated confidence for sample i (the selected class). This is the formulation for a multiclass problem.



T. [1]

How to choose the temperature value T

The question arises ¿How to choose a T that can communicate the appropriate confidences that the model should have on the outputs? Intuitively a model that is 99% confident should, in fact, be correct 99% of the time.

To obtain the best ***T*** the model performs NNL To obtain the best T, the model performs NLL (Negative Log-Likelihood) minimization. On the code this optimization is performed using bounded scalar minimization (Brent's method) over the interval $T \in [0.1,10.0]$, requiring only milliseconds to converge and preserving the model's classification accuracy

Also, the quality of the calibration of a model can be measured by the **Expected Calibration Error (ECE)**, this metric shows how well the predicted probabilities of a classification model are calibrated. It compares the model's average confidence with its actual accuracy across different probability bins.

ARCHITECTURE

The classification model employed in this study is based on the LeNet5 architecture originally proposed by Yann LeCun in "Gradient-Based Learning Applied to Document Recognition". However, to address the limitations of the original 1998 design and improve performance on modern datasets, several key modifications were implemented to create a modernized variant suitable for CIFAR-10 binary classification

Network Structure

The modernized LeNet5 architecture consists of three convolutional blocks followed by a fully connected classifier:

Modern Improvements Implemented

The baseline architecture incorporates several modernization techniques to the LeNet5:

1. **MaxPooling instead of Subsampling**: The original trainable average pooling layers were replaced with standard 2×2 max pooling with stride 2, reducing computational cost while improving feature extraction

2. **ReLU Activations**: Replaced sigmoid/tanh activation functions with ReLU ($\max(0, x)$) to alleviate vanishing gradient problems and accelerate convergence

3. **Kaiming Initialization**: Convolutional layers use Kaiming normal initialization (fan_out mode) specifically designed for ReLU activations, while linear layers employ Gaussian initialization ($\mu = 0, \sigma = 0.01$)

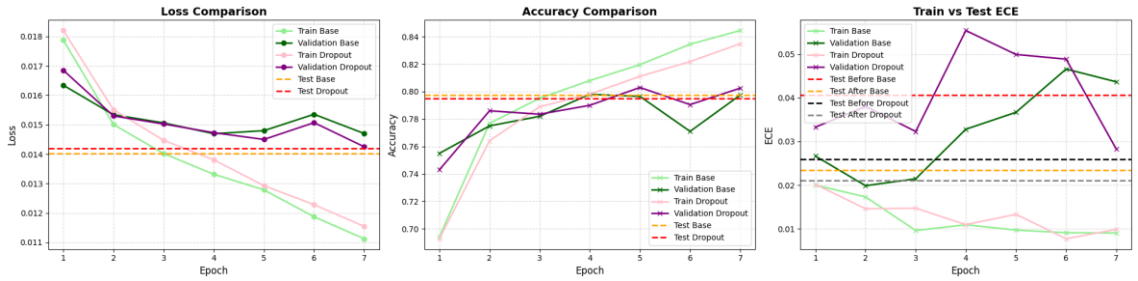
4. **Adam Optimizer**: Utilized Adam optimization with learning rate $\alpha = 0.001$ instead of traditional stochastic gradient descent (SGD) for adaptive learning and faster convergence

5. **LogSoftmax Output**: Applied log-softmax activation at the output layer for numerical stability during cross-entropy loss computation

Regularized Variant

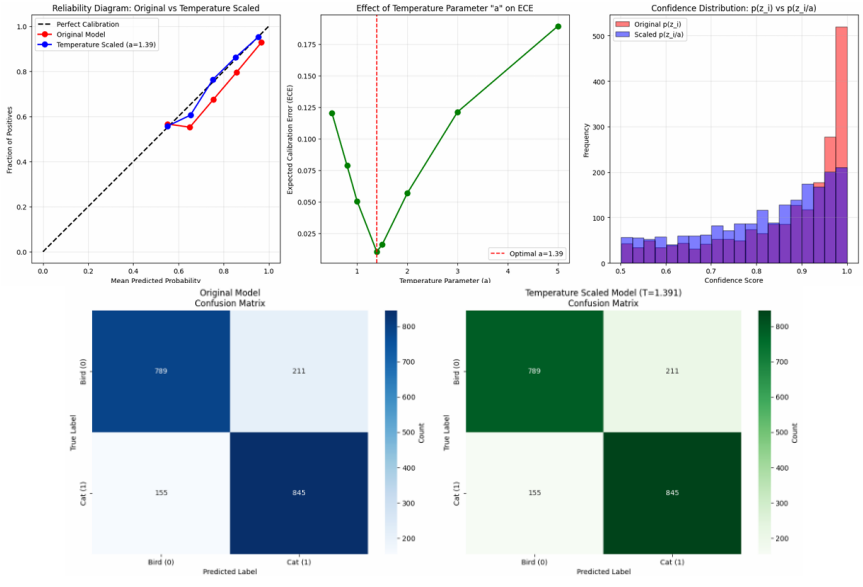
To investigate the effect of regularization on model calibration, a second variant was trained incorporating dropout layers, with a dropout rate of $p = 0.2$:

RESULTS & DISCUSSION



For this binary classification task of distinguishing birds from cats in CIFAR-10, the baseline model without regularization outperforms the dropout-regularized model across most aspects, achieving higher final test accuracy (80.9% vs 79.7%), faster and smoother convergence, better utilization of the 10 training epochs. These results suggest that for the dropout to start working, there is a need of more available data. Also, it's observable as well that the models start plateauing on the 5th epoch, which means that after this epoch the model is just memorizing the dataset and start to overfit.

Regarding the temperature scaling we can also see the ECE results after and before the Temperature scaling are improved from a 4.06% to a 2.35% which is an improvement of a 57%, we can see below the relation of confidence vs accuracy on the graphs below, and how the ECE is minimized vs the different values of T and the confusion matrix of both models



Finally we want to discuss briefley the case of temperature scaling on a big model such is ResNet18 with a final accuracy on this problem of ***88.5%*** wich is better than the LeNet5 but with much bigger ECE values, wich means more overconfident, with a result of ECE of the base model of 3.4% and after apply temperature scaling a 1.7% improving the final accuracy of the model to 97.6% wich is much better than before.