NEURAL NETWORKS: PROJECT 2

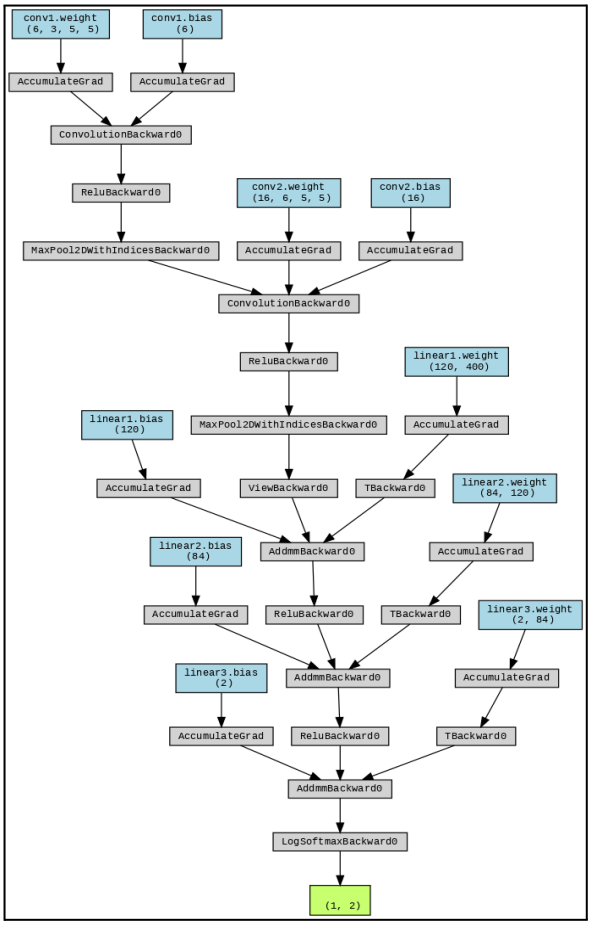
Authors:
Marina Gómez Rey (100472836)
Ángela Durán Pinto (100472766)
María Ángeles Magro Garrote (100472867)

Abstract
This lab investigates the calibration of convolutional neural networks (CNNs) in a classification setting, focusing on classifying birds from cats in the CIFAR-10 dataset To do that, a review of the importance of calibration in neural networks was done, focusing on sections 1 to 4 of the paper "On Calibration of Modern Neural Networks." Subsequently, a Lenet5 CNN is trained from scratch for the bird-cat classification task, and its calibration is evaluated using reliability diagrams and Expected Calibration Error (ECE) on the test set. Basic temperature scaling (Platt's scaling) is implemented to refine the model's output probabilities, obtaining the best temperature for our model and studying the effect of the scaling parameter 'a'. Finally, as an extension, the experiment is repeated with a larger model obtained by fine-tuning a pre-trained model's last classification layer.

Creation of the model
LeNet-5 stands out among other convolutional neural network (CNN) architectures due to its hierarchical structure, featuring alternating convolutional and subsampling layers for effective feature extraction and abstraction. It introduced convolutional and subsampling operations, demonstrating their efficacy in image recognition.

The dimensions chosen in the Lenet5 architecture are carefully selected to ensure effective feature extraction and classification while minimizing computational complexity.

- With a 32x32 input size for CIFAR-10 images, the architecture starts with two convolutional layers applying 5x5 filters to capture low-level features and gradually increasing the number of filters to learn more complex features.
- Max-pooling layers with 2x2 kernels are applied after each convolutional layer to downsample features while retaining important information.
- The fully connected layers (linear1, linear2, and linear3) reduce feature dimensionality. These dimensions start at 400 (16x5x5) after flattening the output of the second convolutional layer, ensuring efficient utilization of extracted features. The subsequent linear layers further reduce dimensionality to 120 and then to 80.
- The calculation of the spatial dimension (final_dim) ensures that downsampling operations maintain spatial information. Starting with the input dimension, it subtracts the filter size (4) to account for convolutional operations and then divides by 2 to represent the downsampling effect of pooling layers. This process is repeated twice, accounting for two convolutional-pooling pairs. In the case of CIFAR-10 images (32x32).

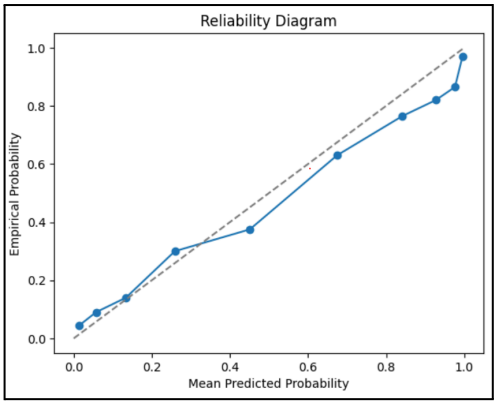


Dimensional structure of Lenet 5 CNN
(extend view to see it)

Furthermore, early stopping was applied to the model for preventing overfitting, with the parameter patience to personalize how flexible the model could be in case of an increase in the validation loss.

Reliability diagram & ECE

In order to evaluate the reliability of our classification model using a reliability diagram and the Expected Calibration Error (ECE). The reliability diagram visually compares the mean predicted probability against the empirical probability, helping us understand the model's calibration. We implement this diagram using the *plot_reliability_diagram* function, which plots the probabilities and a diagonal line representing perfect calibration. Additionally, we compute the ECE, a scalar measure of calibration error. The *ece_cal* function calculates the ECE by partitioning the predicted probabilities into bins and comparing them to the true probabilities. A lower ECE indicates better calibration, with perfect calibration yielding an ECE of 0.



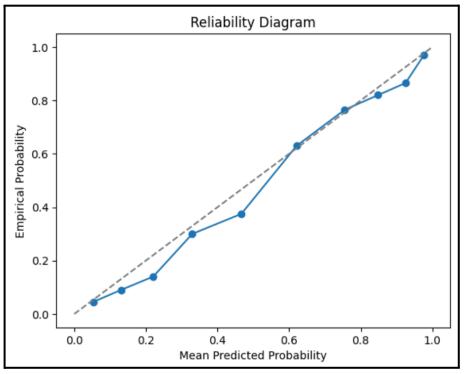
Reliability diagram w/o calibration of Lenet5 CNN model (ECE = 0.055)

Calibration

Implementing temperature scaling, also known as Platt's scaling, is a form of calibration for neural networks. Temperature scaling adjusts the output probabilities of the model to better align with the true probabilities. By applying a scaling factor *a* to the logits *zi* , where *zi* represents the raw output of the network before softmax, we can adjust the sharpness of the predicted probabilities. This scaling helps to improve the calibration of the model, ensuring that the predicted probabilities reflect more accurately the confidence of the model in its predictions.

In order to do this, two approaches were taken:

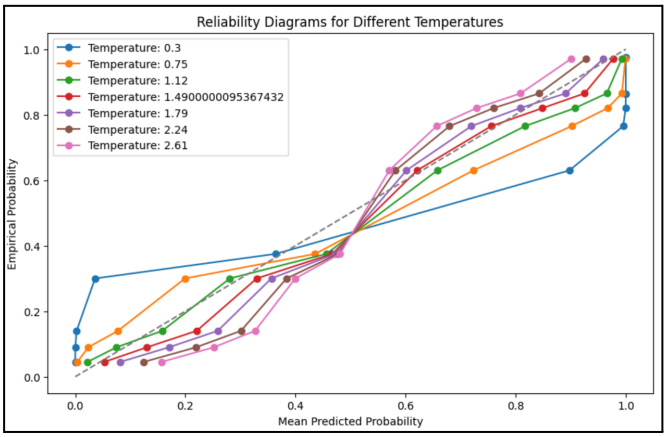
1) Optimizing Temperature: Initially, the temperature parameter *a* is optimized using a validation set. This involves training the neural network and adjusting the temperature to minimize a calibration error metric such as the expected calibration error (ECE). The goal is to find the value of *a* that aligns the model's predicted probabilities with the true probabilities. The NLL was compared before (0.449) and after (0.429) obtaining the optimized temperature:



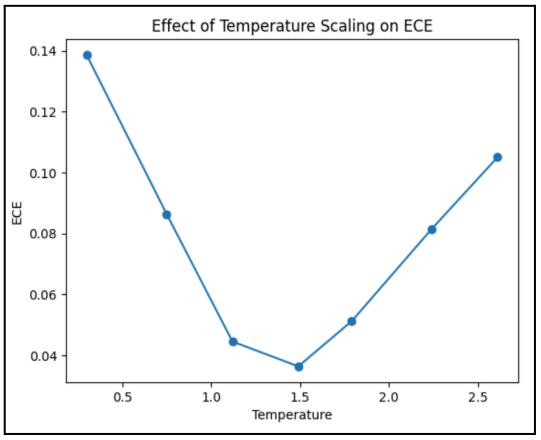
Reliability diagram of calibrated Lenet5 CNN model (ECE = 0.036)

2) Testing with Several Values: After obtaining the optimal temperature through optimization, the model is tested with several values of *a* to validate that the optimized temperature indeed yields the best calibration performance. This step ensures that the selected temperature effectively adjusts the sharpness of the predicted probabilities and improves the overall calibration of the model.

Due to the calibration, the reliability diagram of the model with the optimized temperature is closer to the calibration line, and in fact, is the one with a better ECE.



Reliability diagram depending on temperature



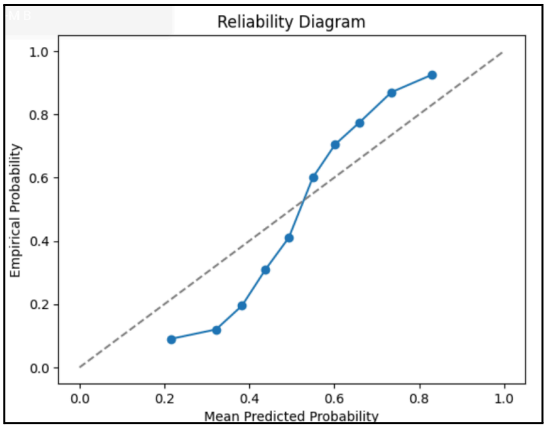
ECE depending on temperature

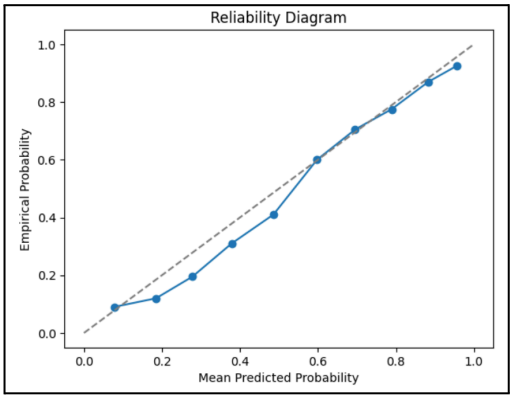
Bigger model

In this second model, we are fine-tuning a pre-trained DenseNet model for the CIFAR-10 dataset, which has different input image size and classes from the original dataset. Therefore, we need to resize the input images to match the model's expected size (224x224) and apply the same normalization. Additionally, we use Data augmentation techniques in the training dataset to improve the model robustness. Then, we replace the final classifier layer to fit our task of classifying birds and cats. During training, we freeze the pre-trained convolutional layers and only update the new classifiers' parameters, allowing transfer learning.

As in the first part, we have tried our model for both the non-calibrated and calibrated cases. However, the best temperature has been obtained by trying several values that were stored in an array.

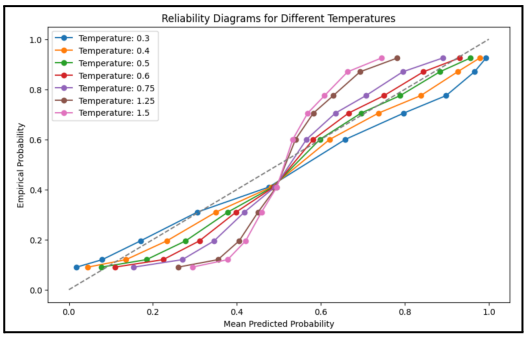
The results that have been obtained after running the code have been the following:

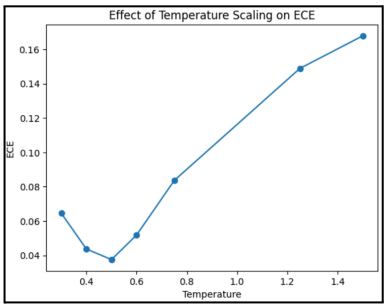




Result without calibrating (ECE: 0.122) *Result with calibration (Temperature: 0.5, ECE: 0.037)*

As a summary of the results of the temperature iteration, the following graphs have been obtained:





Reliability comparison with diff. temperatures *Effect on ECE of the temperatures*

An important observation highlighted is the initial calibration disparity between the LeNet and DenseNet models. While the LeNet model begins with superior calibration, DenseNet initially yields poorer results, indicating a **greater need** for calibration refinement. Moreover, it must be stated the **importance of optimizing the temperature** parameter rather than employing random values. Failure to do so may lead to overlooking the most effective configuration.