Neural Networks

Carlos III University of Madrid

April 08, 2024

Practice II – Understanding calibration in CNNs.

Pablo Gradolph Oliva - 100458456

Raquel Parajuá Delgado - 100454359

TABLE OF CONTENTS

1. **Introduction**
2. **Results**
    - 2.1. **Reliability Diagram and ECE without Platt Scaling.**
    - 2.2. **Reliability Diagram, ECE and study of parameter 'a' with Platt Scaling.**
3. **Optional part. Pre-trained model.**
4. **Conclusions.**

1. Introduction.

In this second project we've evaluated calibration in convolutional neural networks (CNNs) for a classification setting. The goal of calibration is to ensure that the estimated class probabilities are representative of the true correctness likelihood, this is to produce probabilities reflecting the reality. Although this has been a very strong field in computer vision tasks, problems related with the complexity and huge capacity of models and the increase of miscalibration have arisen, making it still an important area of research.

Reading the paper "*On Calibration of Modern Neural Networks"* we've seen what factors may influence calibration: such as depth, width, weight decay, and Batch Normalization, showing for example that while very deep/wide models are able to generalize better and easily fit the training set, they negatively affect model calibration. Also, that less weight decay means more miscalibration or that models trained with Batch Normalization slightly improved accuracy but were worse calibrated. In the end, simple models and techniques (Temperature scaling) have shown to outperform other more complex methods and obtain a better result when it comes achieving a better calibration.

To analyze this by our own means, we've trained a CNN model (Lenet5) from scratch, to classify bird from cat images within the CIFAR10 assemble and used as a visualization tool the corresponding Reliability diagram and the Expected Calibration Error (ECE) as a numerical value.

Then, following the findings mentioned in the paper, Temperature Scaling was applied to improve the calibration of our model. Temperature Scaling is a simple extension of Platt Scaling, which is a parametric, post-processing approach that adjusts the output probabilities by modifying the logits before applying the SoftMax function. In Temperature Scaling we use a single scalar parameter 'a' whose effect is going to be studied to understand how its change can have an impact on our model calibration.

Finally, we extended our research by applying these concepts to a larger pre-trained model, by finely tuning its last classification layer.

If for some reason, you cannot see the images in the attached Jupyter Notebook, please access this GitHub repository to see them: https://github.com/PabloGradolph/Neural-Networks/tree/main/Project%202

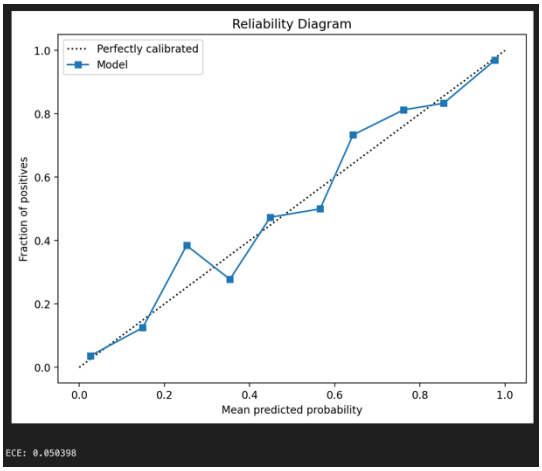
2. Results

2.1 Reliability Diagram and ECE without Platt Scaling.

We have a binary classification problem with two classes, which will be labeled as positive (1) and negative (0). For each prediction made by our model, we will have a set of probabilities for each class and since it is a binary problem, the sum of the probabilities of both classes must be 1.

When calculating the Reliability Diagram and ECE, we evaluate how well calibrated the predicted probabilities of our model are compared to the true labels. If the probability of the positive class is well calibrated, by extension, the probability of the negative class will also be calibrated, since the two are intrinsically related. In other words, a good calibration in the predictions of the positive class will be reflected in the negative class due to their complementary relationship. This tells us that in a binary problem the analysis of

the calibration of one class is enough to understand the calibration of the whole model, since you cannot have one class well calibrated and not the other one in this binary context.



In this Reliability diagram before applying Platt Scaling, we can distinguish two lines:

- The dotted line represents perfect calibration. In a perfectly calibrated model, the predicted probability matches the true frequency. For example, if a model predicts a class with a 75% probability, then we would expect that class to be true 75% of the time.
- The blue solid line shows the calibration performance of our model. Take for example the first dot on the left, it may represent all the predictions where our model was 10% sure that the image was a cat. If that point is above the dotted line, it means the model is underestimating the probability; if it is below, it is overestimating.

Overall, we can see that the model might need some improvement on calibration even though it is not extremely miscalibrated.

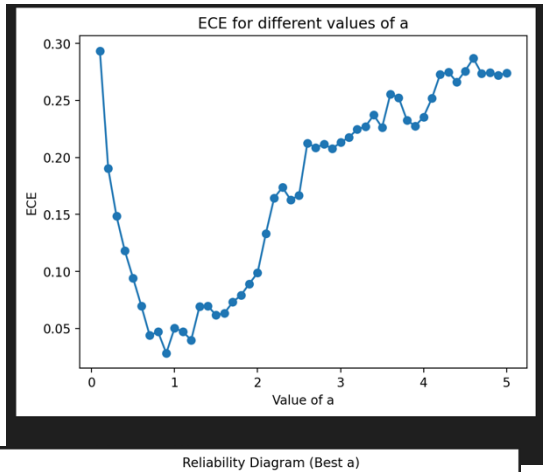
The ECE is a measure that summarizes the difference between the predicted and observed probabilities. For a perfectly calibrated model the ECE is 0, so we want a value as close to 0 as possible, meaning we have a well calibrated model. An ECE of 0.050398 indicates that, on average, the difference between the predicted and observed probability is about 5%, which is not that bad, but we want to study if applying Platt Scaling this value decreases even more and we obtain a better calibrated model.

2.2. Reliability Diagram, ECE and study of parameter ‘a’ with Platt Scaling.

When applying Platt Scaling, specifically Temperature Scaling, the objective is to find a “temperature” factor to “smooth” the model’s predicted probabilities. This factor adjusts the scale of the logits before applying SoftMax to obtain the new fitted probability.

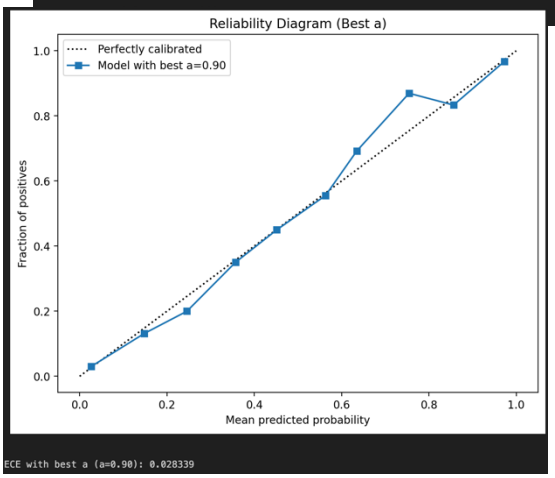
This process finds the temperature parameter ‘a’ that best calibrates our model and then applies this setting to the logits before running them through SoftMax to obtain the calibrated probabilities.

We first evaluated Temperature scaling for a value of a=1 with which no change in the Reliability Diagram or on the ECE is observed, which makes sense since we don’t change the logits .



We then evaluated variations in the ECE according to the value of 'a' between 0 and 5, negative values don't maake sense. Then we keep the best value of 'a' obtained and see how our diagram and ECE changes.

The best value of 'a' is a = 0.9 which corresponds to an ECE of 0.028339 (improving from the one in the 2.1 section).



And the Reliability Diagram obtained with this value of 'a', as can be visually appreciated, improves the one from the previous section, indicating a better calibration of our model.

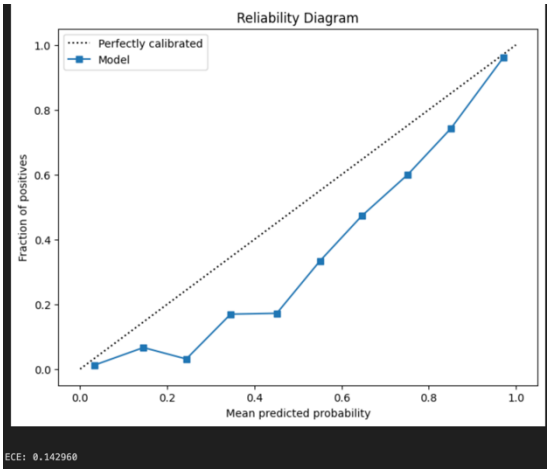
3. Optional part. Pre-trained model

We will use a pre-trained model provided by PyTorch, such as ResNet, for example. This process involves loading a pre-trained model, replacing its last classification layer to fit your specific task (in this case, classifying between birds and cats), and then training only this last layer.
When we use a pre-trained model, such as ResNet, on specific tasks other than those it was originally trained for, we generally follow a fine-tuning process, which involves two main steps:

- Transfer Learning: We load the pre-trained model and optionally freeze the weights of some or all of the previous layers so that they are not updated during training. This is useful because the initial layers of a vision model capture general features (such as edges, shapes, etc.) that are applicable to a wide variety of computer vision tasks.
- Fine-Tuning: We tune the last layers of the model (or the entire model, depending on the chosen strategy) to our specific data set. This allows the model to adapt its weights to the particularities of our specific problem.

Although the model comes pre-trained (i.e., it has already learned a set of useful features from a large data set like ImageNet), fine-tuning is necessary because the data set and the specific task (bird vs. cat classification) may differ significantly from the original 1000-class classification task for which it was trained. By training (or retraining) the model on our specific data set, we allow it to tune its parameters to improve its performance on our specific task.

The process is the same as in the first part but with a ResNet50 model, which is substantially more complex than ours.



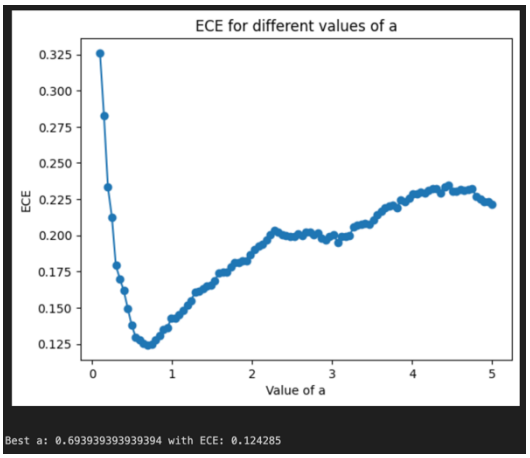
Calibration in this model, as can be easily observed, is far more off than in our model, and the ECE value of 0.142960 indicates that the difference between the predicted and observed probability is about 14.3%. This event matches with the predictions made in the article, where it was stated that more capacity and complexity of the models meant more miscalibration.

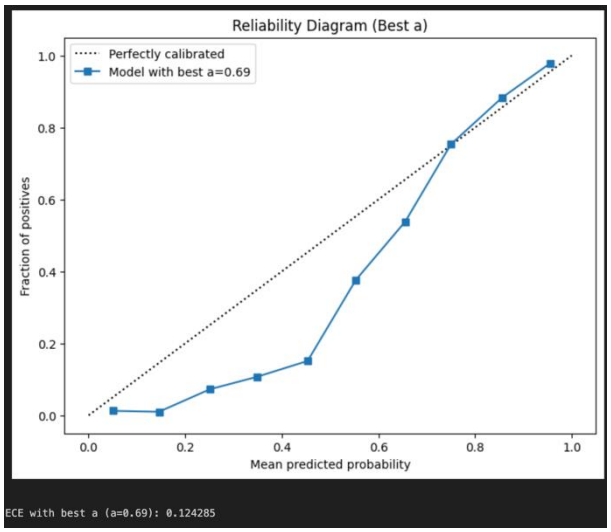
After this, Platt Scaling was applied, specifically Temperature Scaling as before, and it was first evaluated for a value of parameter a=1. As it was previously seen and stated this didn't involve a change neither on the Reliability Diagram or on the ECE value.

Then, we searched again for the best value of 'a' and take the one with lowest ECE to introduce it in our model.

In this case, the best value was for a=0.6939 with an ECE of 0.124285.

Then this value was used to plot the corresponding Reliability Diagram and the following was observed:





We can say form this visualization tool that miscalibration is improved slightly and this is also what the value of the ECE tells us. However comparing it to the calibration of our own model, this one is reasonably worse, effectively corroborating the observations made in the paper that showed that more complex models tended to be more miscalibrated than simpler ones.

4. Conclusions.

Overall, first reading the paper and then evaluating the statements made by our own means has made it easier to understand how calibration works in CNNs. We have checked that for simpler models as our own, calibration was quite better than the one obtained for a more complex model as the ResNet50 one, as smaller ECE was obtained and the Reliability Diagram represented this fact, and this was exactly what was shown in "*On Calibration of Modern Neural Networks*". It remains for future research to understand why this happens and

why even though accuracy is improved calibration is negatively affected.

Nevertheless, we've also seen how simple calibration techniques such as Temperature Scaling can effectively correct the miscalibration phenomenon in neural networks, improving miscalibration in each case independently of the model's capacity.