BAYESIAN DEEP LEARNING CLASSIFICATION ON NON-CURATED DATASETS

Alessandro Baldo
Data Science and Engineering
alessandro.baldo@eurecom.fr
github.com/alessandrobaldo/BayesianDeepLearningOnNonCuratedDatasets.git

February 6, 2021

ABSTRACT

Bayesian Neural Networks represent a different approach to deep learning, promising a robust tuning of the model's parameters, treating the entire problem as Bayesian inference and taking into account the true posterior distribution of such parameters with respect to data. These techniques are often supported by sampling methods, as a framework for the update of weights and biases. In this paper, these methods are used to measure the impact of scaling the true posterior distributions. The experiments will be held over datasets characterized by different levels of curation.

1 Introduction

Recently, the Bayesian treatment of deep learning has been the center of the attention of some studies [1], [2], which presented extensive experimental campaigns, demonstrating the effect of adjusting the posterior distribution. In [1], the concept of "cold" posteriors was applied in well-noted deep models for image classification, and their benefits, combined with the robustness of Stochastic Gradient Monte Carlo methods [3], led to an average increase of the accuracy.
In [2], the concept of "cold" posterior was then questioned and compared to a slightly altered version: the "tempered" posterior; if in the former, the posterior is entirely scaled by a temperature parameter T (1), resulting in a narrower space, the latter consists only in appropriately scaling the likelihood term, supporting the thesis of the weak effects of priors over the model (2).

$$\log p(\theta|\mathcal{D}) = \frac{1}{T}\log p(\mathcal{D}|\theta) + \frac{1}{T}\log p(\theta) \tag{1}$$

$$\log p(\theta|\mathcal{D}) = \frac{1}{\lambda}\log p(\mathcal{D}|\theta) + \log p(\theta) \tag{2}$$

Furthermore, [2] argued the effectiveness and the relevance of such improvements presented by [1], since the experiments were limited to models applied on curated datasets, where the ambiguity on the labels is very reduced.
Starting from this recent debate, the ultimate scope of this paper is to validate or not the latter hypotheses, contextualizing the models over non-curated datasets.

2 Overview of Bayesian Statistics

Bayesian inference is a branch of statistics, whose core is stated by the Bayes' Theorem:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \tag{3}$$

This quite simple rule defines the so called *posterior distribution* over model parameters as depending on a *likelihood* term $p(\mathcal{D}|\theta)$, representing the observed distribution of data conditioned to the model parameters and a *prior* term $p(\theta)$ as a summary of the initial beliefs on the model parameters. Finally, the denominator represents the marginal likelihood, a common term representing a weighted average of all the possible models included in the prior distribution.
Given the posterior distribution, we can make predictions about new instances x, by averaging over all the likely models, defining the so called *posterior predictive* (4).

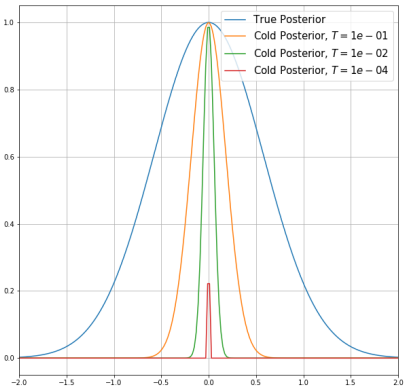
$$p(y|x,\mathcal{D}) = \int p(y|x,\theta)p(\theta|\mathcal{D})d\theta \tag{4}$$

In fact, Bayesian statistics does not aim for the optimization of a single model, but it instead takes into account a variety of models by opportunely weighing them.
In general, the knowledge about a posterior distribution can be summarized with the *MAP* (Maximum a posterior) (5). That is a point-wise estimator, denoting the maximum/mode of the posterior and it is especially informative in case of uni-modal distributions.

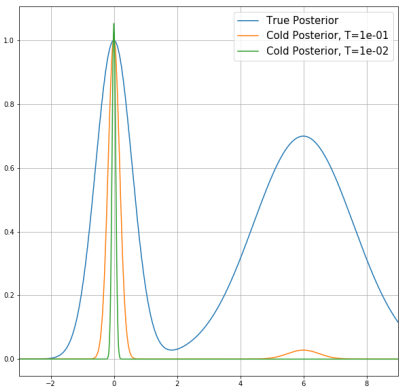
$$\hat{\theta}^{\mathrm{MAP}} = \underset{\theta}{\operatorname{argmax}}\, p(\theta|\mathcal{D}) \tag{5}$$

2.1 Intuition of Scaled Posteriors

As briefly discussed in the Introduction, scaled posterior distributions are governed by a *temperature* parameter (1,2), or scaling factor, which is generally chosen such that the overall distribution results narrower and the density of information is still more concentrated around the MAP estimate (figure 1a). We could understand well, that such an operation takes along some risks, especially whether the posterior distribution is multi-modal (figure 1b).



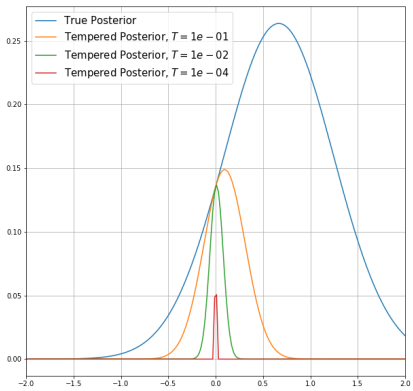
(a) Example of a 2-D Uni-modal Posterior Distribution ($e^{-x^2-\frac{1}{2}x^2}$) scaled with different temperatures



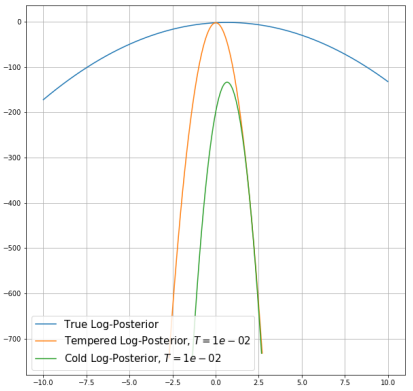
(b) Example of a 2-D Bimodal Posterior Distribution ($e^{-x^2-\frac{1}{2}x^2} + 0.7e^{-\frac{(x-6)^2}{5}}$) scaled with different temperatures

Figure 1: Cold posteriors

As the temperature parameter decreases, the set of samples derived from that posterior is more constrained, and, as a consequence, the ability of a model to cope with uncertainty is shrinked too.
From a more practical point of view, this operation has the role of a more tailored data augmentation, since each data point, in the final classification, would end up to be overcounted. In this way, the result is an overall amplification of the proportions between certainty and uncertainty referred to the dataset, which leads to a properly different model with respect to the original one.
As [1] highlighted, there could be some circumstances where customized versions of the posterior distributions could make achieve higher accuracies, benefitting of the low levels of uncertainty in the datasets.



(a) Example of a 2-D Uni-modal Posterior Distribution ($e^{-x^2-\frac{1}{2}(x-2)^2}$) scaled with different temperatures



(b) Comparison of Log-Posteriors ($e^{-x^2-\frac{1}{2}(x-2)^2}$): it is evident the shrinking effect of the temperature scaling in both cases and the shift of the MAP estimate of the Tempered Posterior

Figure 2: Tempered posteriors

Recalling Equation (2), the tempered posterior only modifies the likelihood term. From an analytical point of view, the difference with the cold posterior is in the evolution of the position of the MAP estimate: if the cold scaling has the effect of still more accumulating information around

it, the tempered provokes a shift too, majorly differentiating the resulting model (figure 2).

2.2 Deep Learning Context

Deep Learning generally deals with unconstrained non-convex problems, by trying to solve an optimization problem, whose objective is the minimization of a *loss function*:

$$\min_{\theta} L(\theta, D) \tag{6}$$

In this form, it points out that the scope is to find the optimal set of parameters θ, such that the loss can be minimized. Of course, this loss is strictly dependent on the data the model is fed of.
Expanding this comparison, to the previous introduction of Bayesian statistics, it is easy to map this argument of the objective function into a likelihood term.
Regularized Neural Networks look beyond that and consider a regularized version of this loss function, taking the form:

$$\min_{\theta} L(\theta, D) + \lambda||\theta|| \tag{7}$$

This new addend has the effect to favour simpler models, since it introduces a bias limiting the values parameters could take.
Bayesian treatment of deep learning includes this concept, adopting the so-called *negative log-prior* term (8). By encoding the prior beliefs, it ensures also to (L2-)regularize the entire network and thus keeping the variance of the parameters in a restricted range of values.

$$\frac{1}{2}\sum_{i} w_i^2 \tag{8}$$

3 Sampling Methods

Sampling methods are widely used techniques which allow to draw samples from target distributions, which would be not analytically tractable for non-linear models, as deep learning deals with. Then, their effectiveness, together with a suitable estimation of the confidence intervals, make the sampling methods a quite adopted tool in Bayesian Neural Networks.
In this paper, the focus was mainly on Metropolis and Markov Chain Monte Carlo (MCMC) algorithms.

3.1 Metropolis

The Metropolis algorithm is a sampling method, guaranteeing the convergence to the true posterior when finely tuned. Our scope is to sample from the true posterior p, but since it is not known, a distribution f and a symmetric proposal distribution $J(x^*|x_i)$ are taken into consideration. The former must be such that $f \propto p$ and will coincide with the loss function defined at the deep learning model level.
The iterative procedure provides for a simple update of the set of parameters, that could be accepted or not, by sampling from the proposal distribution (9).

$$\theta_i \leftarrow \theta_i + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{9}$$

Given the step i, a new set of parameters can be either directly accepted or not. The direct acceptance happens in all those cases where the regularized negative log-likelihood (i.e. f^*) at that step results lower than the related loss at the step $i-1$.
If this does not happen, the sample can be still conditionally accepted if the following statement is true:

$$r < \frac{f^*}{f_{i-1}}, \quad r \sim \text{Uniform}(0,1) \tag{10}$$

Otherwise, the updated parameters set is rejected and the previous one in maintained.
Given n iterations, the Metropolis algorithm allows to collect n parameters sets. We define an acceptance ratio, as an indicator of diversity measure among all the sets.
Finally, the last step of the algorithm consists of disregarding the outcomes of the first iterations (since the random walk would not surely be converged to the target posterior distribution) and then considering only a subset of non-neighbouring parameter sets, in order to completely minimize the correlation between contiguous results.
The model is then evaluated as an averaging over the sets the algorithm kept.
In general, a more generalized version of this algorithm is taken into account, namely the Metropolis-Hastings algorithm. It allows to expand the application of such a method also to non-symmetric proposal distributions. The overall algorithm differs only for the evaluation of the loss in the acceptance rule (11).

$$r < \frac{J(x_i|x^*)}{J(x^*|x_i)}\frac{f^*}{f_{i-1}}, \quad r \sim \text{Uniform}(0,1) \tag{11}$$

3.2 Markov Chain Monte Carlo (MCMC)

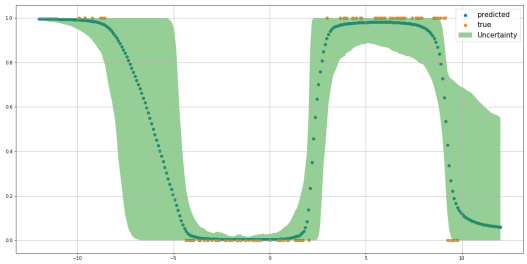
The family of Markov Chain Monte Carlo algorithms [4],[3] are very known for their effectiveness in the sampling field. Their nature allows to better direct the random walk nature, leading to more consistent and concrete results.
Hamiltonian Monte Carlo (HMC) is a method providing high acceptance ratios from distant proposal distributions, by incorporating a Metropolis-Hastings procedure to define the acceptance/rejection rule. It deals with a contextualization of the parameters in a physical system scenario. The log-posterior distribution is associated to a *potential energy term* $U(\theta)$ and it is plugged into the aforementioned MH framework, to define more concrete distributions than random-walk proposals.
A second physical measure determines the total energy of the system: the *kinetic energy*. This term is defined by a set of *momentum* variables r and a mass matrix M (12).

$$K(r) = \frac{1}{2} r^T M^{-1} r \tag{12}$$

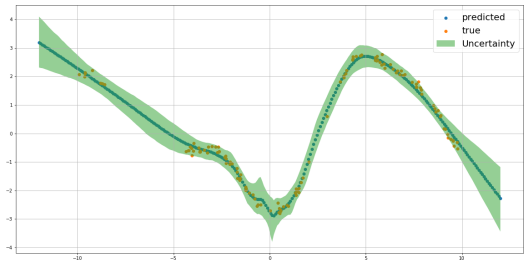
The final measure defining the state of the system is the so called *Hamiltonian function*, which results into the discriminative measure used in the acceptance rule at the end of each iteration.
Starting from this definition, the Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) allows a more efficient computation of the gradients, which would be massively costly in large model instances. As Stochastic Gradient aims to generalize by computing gradients only on a small fraction of the entire dataset, the SGHMC includes the concept of batches $\tilde{\mathcal{D}} \subset \mathcal{D}$, over which the update procedure is performed (13).

$$\nabla \tilde{U}(\theta) = \frac{|\mathcal{D}|}{|\tilde{\mathcal{D}}|} \sum_{x \in \tilde{\mathcal{D}}} \nabla \log p(x|\theta) - \nabla \log p(\theta) \qquad (13)$$

It then defines a matrix B, which takes into account the contribution of the gradient noise introduced by the batches and, as a further improvement, a *friction* term: without it, the SGHMC would usually require to compute the costly Metropolis-Hastings step, or, alternatively, long runs with low acceptance probabilities. The direct consequence is that in the end such method presents a quasi-completed decorrelation between consecutive samples, so there is no need to still separate correlated parameters at the end of the procedure. Normally, the algorithm itself requires an initial number of *burn in* iterations [4], whose function is to only put the system in the most suitable conditions.



(a) Metropolis algorithm on a simple classification problem



(b) Metropolis algorithm on a simple regression problem

Figure 3: Metropolis algorithm on two simple instances

4 Numerical Experiments

4.1 ANNs on Low-dimensional datasets

A basic ANN was built, consisting of one 50-neurons hidden layer. Such structure has been applied to two simple tasks, operating on a 1-dimensional dataset, presenting many missing values at the extremes and in the central region. The two tasks involved a regression and a binary classification procedures, respectively.
The loss function was suited to the Bayesian scenario, as stated in (7), by respectively choosing a *MSE* loss for the regression problem and a *Binary Cross-Entropy* for the classification of the points, as the negative log-likelihood term. In both cases, instead, the prior assumption was materialized through a *L2-Regularization* (8).
Metropolis algorithm has been tested for long runs on both the two problems, denoting an effective behaviour in updating model parameters. As the two figures 3a and 3b report, the confidence interval correctly reflects the missing data zones, being instead very narrow whether the uncertainty resulted low.
For this representation, the 25% rule on the acceptance ratio was respected, being a well-established heuristic approach to control variability of samples.
The entire set of parameters was then appropriately segmented, discarding the first few thousands of iterations and considering the samples with a step size of a hundred units, since, a correlation bias would exist between consecutive samples.

The numerical experiments on the small networks then pursued by testing the SGHMC counterpart. Due to the limited size of data, these trials did not take into account the stochastic nature by using mini-batches.
As previously discussed, the nature of Stochastic Gradient HMC allows to better characterize the random walk nature, by establishing a proper direction for the updates. In practice, this results in a method leading faster to convergence, in average. The SGHMC test presented a major sensitivity on those areas where data was sparse (figure 4), not completely evidencing the real existing gap in performances with respect to the Metropolis algorithm.

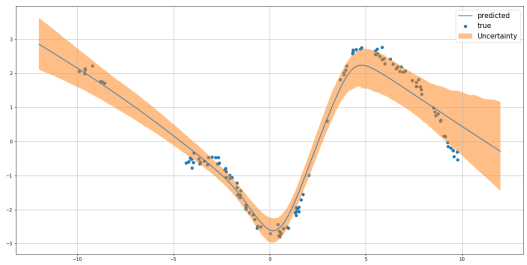


Figure 4: SGHMC on a simple regression problem

4.2 ResNet20 on CIFAR-10

CIFAR-10 represents a standard curated dataset, involving 60'000 images divided in 10 different classes. The organization of the tests initially looked at the behaviour of the Metropolis algorithm, which, however, had to cope with the large number of model parameters. Its random walk nature proved to not be adequate for this complex image classification task. Using SGHMC provided instead the optimal trade-off with the model complexity. In particular, that is achieved by feeding the network with small instances of the training set, namely the mini-batches.
The theory behind highly relies on the central limit theorem, considering samples as independent [3], and leading to a correct measure of batch size not smaller than some hundreds of data points: it is then evident the reduction of computational cost, with respect to compute gradients on an entire training set.

4.2.1 Cold Posterior

The experiments tried to replicate what was held in [1], by measuring the increment of the accuracy when adopting cold posterior distributions. Figure 5 highlights how temperature scaling on both the likelihood and prior terms has an improving effect, picking the best five models for baseline accuracies (i.e. $T = 1$). However, by considering different set of hyper-parameters, corresponding to low performing models, there is some numerical evidence of a worsening of the performances at larger scaling factors, especially whether the learning rates got larger.
This assumption is quite coherent, since, as the temperature scaling enlarges the loss value to be backpropagated, a too big learning rate would provoke similar overwhelming effects on model's parameters' updates, leading to a conflicting configuration.

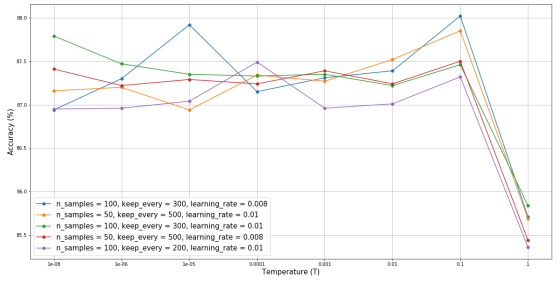


Figure 5: Cold Posterior effects on curated CIFAR-10

The simulation of non-curated datasets required to gradually increase the noise in CIFAR-10. This was done, testing separately the noise effects when flipping a percentage of the labels and directly corrupting the images by introducing some Gaussian noise.
The main observed and expected consequence was an average reduction of the baseline model's accuracy. However, the application of a cold version of posterior, still led to a proportional improvement of the metric. It is difficult to say whether the assumption of [2] on non-curated datasets could be wrong or not, however it is remarkable that the artificial noise introduced on the labels (figure 6) seems to perturb less the predictive capabilities of the model, if compared to the symmetric case on the images (figure 7). In the latter case we particularly assist to a general flattening on the beneficial effects of temperature scaling, suggesting that the sensitivity of the Bayesian framework towards data points could be higher with respect to classes.

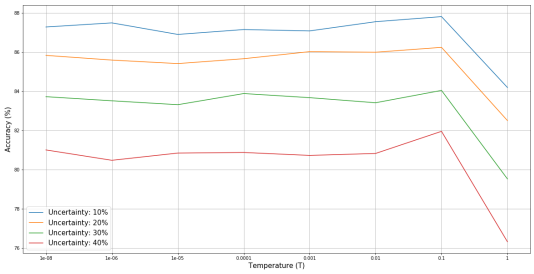


Figure 6: Cold Posterior effects for different levels of non curation on CIFAR-10's labels

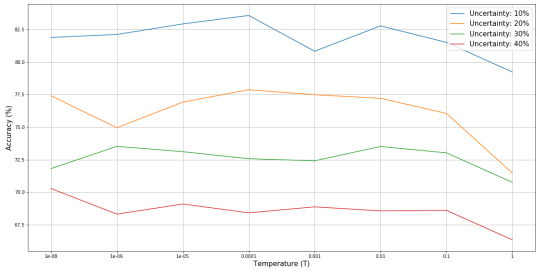


Figure 7: Cold Posterior effects for different levels of non curation on CIFAR-10's images

4.2.2 Tempered Posterior

The tempered version exposed a singular behaviour compared to the cold scaling. Indeed, by considering the best models for baseline accuracy, the temperature scaling partially resulted in a decrease of the predictive performances (figure 8).
The second relevant effect was observed by applying the same temperature scaling on low-performing models, characterized by a large step size (i.e. learning rate) and tested on short simulations. Here, the MCMC led to large improvements, often doubling the score metric of the baseline model (figure 9).
The complex scenario does not allow to determine a particular reason for that, but it surely represents the evi-

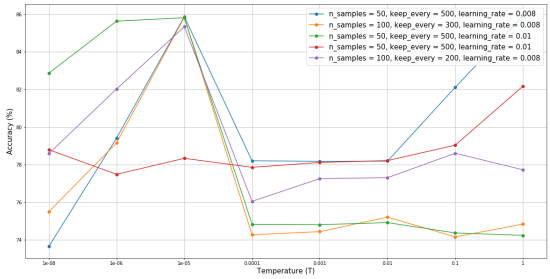


Figure 8: Tempered Posterior effects on curated CIFAR-10: model with top baseline accuracy

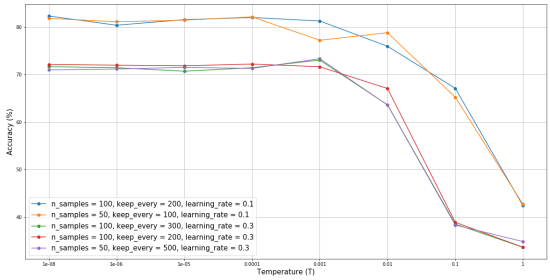


Figure 9: Tempered Posterior effects on curated CIFAR-10: models with low baseline accuracy

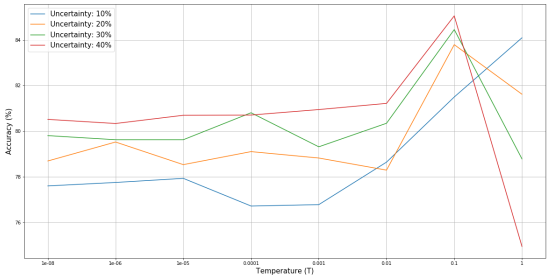


Figure 10: Tempered Posterior effects for different levels of non curation on CIFAR-10's labels

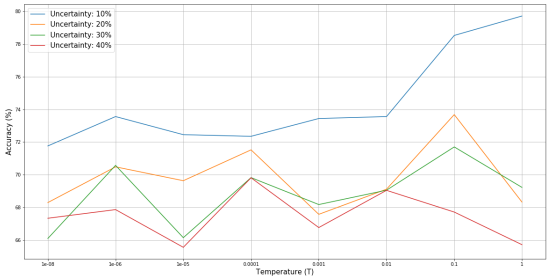


Figure 11: Tempered Posterior effects for different levels of non curation on CIFAR-10's images

dence of how temperature scaling provokes a shift to a completely different posterior distribution. The SGHMC framework surely helped to reach this sub-optimal configuration, covering a "long" distance, due to the large steps. As for the cold posteriors, the labels and images underwent to a corruption process, leading to conflicting results. The experiments introducing noise on the labels pass a wrong message, since performances tend to improve at increasing levels of uncertainty, helped by tempered scaling. The consistency of these results is highly questionable, thus further investigations should focus on how model parameters are affected by noisy labels, which should lead by definition to a drop of the accuracy.
On the contrary, the results obtained by corrupting data points are coherent both internally and with the aforementioned case on cold posteriors. As the uncertainty becomes dominant (i.e. more images become hardly recognizable), the temperature scaling loses all its benefits, reducing to random fluctuations more than real improvements.

4.3 ANNs on Online News Popularity (UCI Dataset)

The Online News Popularity dataset [5] is a large collection of online news binary classified according to the number of times they were shared over Internet. Each news is described by a set of 58 numerical attributes, mostly dealing with the semantic, syntactic and sentiment analyses of the content.
A feed-forward neural network with a retained depth was tested over it, consolidating the Bayesian framework with cold and tempered posteriors. This kind of experiment tried to highlight how simpler data distributions and quite restricted models could have related with temperature scaling and in general with the Bayesian optimization.
The highest baseline accuracy ever reached with this model is around 67%, suggesting that either the features are probably not so informative or the classification task takes along some uncertainty. Using a Cold Posterior was not so beneficial, since the model did not encounter any

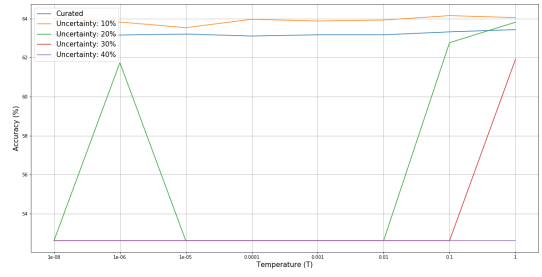


Figure 12: Cold Posterior effects on curated and non-curated scenarios: Online News Popularity dataset

visible improvement. This could be a confirmation of how temperature scaling could be relevant with very deep and complex models, as ResNet-20 is. Here, the non-curation was only simulated by changing opportunely the labels. As figure 12 denotes, the introduction of uncertainty leads the model to become a simple random classifier, thus further considerations are not required.
On the other hand, the tempered scaling recalls the previously mentioned contradictory outcomes, leading to unexpected improvements, especially whether the noise introduction on the classes was heavy.

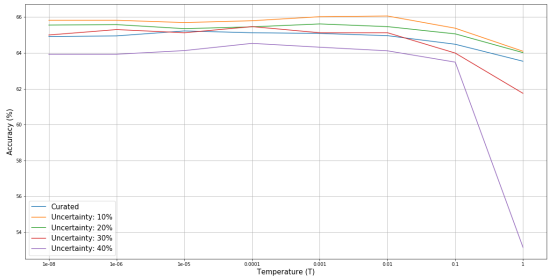


Figure 13: Tempered Posterior effects on curated and non-curated scenarios: Online News Popularity dataset

5 Conclusions

The project tried to retrace the experiments of [1], by enlarging the spectrum also to non-curated datasets, as suggested by [2]. The theories about weak effects of the prior distributions are widely arguable, since the repeated experiments under same conditions of cold and tempered posteriors denoted quite different, and sometims conflicting, results. Indeed, the two techniques lead to different inner properties of the scaled distribution, especially in terms of how the point-wise estimates are moved in the distribution space. Due to this, the tempered scaling proved to be more unpredictable than cold posteriors, which only define a framework coping differently with uncertainty.
In addition, when dealing with CIFAR-10, both the solutions registered a marginal, but existing, decay of the performances only whether the uncertainty directly hit data points. This is in a way coherent with how uncertainty behind a labelled dataset is conceived: being under the supervision of a human knowledge, an uncertain labelling of data would happen for those images of difficult interpretation or whether some biases influence the recognition process. As a consequence, it is hard to say whether simpler datasets (leading necessarily to simpler models) could be really affected and questionned in this kind of Bayesian treatment.
Further studies should thus focus on the (very) deep models counterpart, in order to definitely solve the conflicting natures of tempered and cold posteriors and, thus, to address the effective role of the prior distribution in them.

References

[1] F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin, "How good is the bayes posterior in deep neural networks really?," 2020.

[2] L. Aitchison, "A statistical theory of cold posteriors in deep neural networks," 2020.

[3] T. Chen, E. B. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," 2014.

[4] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust bayesian neural networks," vol. 29, pp. 4134–4142, 2016.

[5] K. Fernandes, P. Vinagre, and P. Cortez, "A proactive intelligent decision support system for predicting the popularity of online news," 2015.