Temperature Scaling

&

Modesty Loss

1 Temperature Scaling

At the beginning of this internship, post-processing approaches were dismissed given that they require a validation dataset to bring calibration. They are assumed to be less powerful than end-to-end learning. Indeed, using the validation dataset may distort the results: to have a correct estimation of the performances of a network, it is preferable to keep the validation dataset unseen by the network.

In addition, we conjectured about the loss of calibration properties when the network is submitted to data from another distribution. Pictures from train and validation datasets are dragged from the same probability distribution. The deployment of the neural network in real life application imply a modification of this distribution, so we assumed that post-processing approaches may not be robust against these modifications.

1.1 Current knowledge

Temperature scaling is introduced in [1] as an extension of Platt scaling for multi-classifiers. This procedure consists in rescaling scores before transposing them into the probabilistic space. To do so, softmax function is modified to integrate a temperature parameter:

$$\boldsymbol{\sigma} = softmax\left(\frac{\boldsymbol{h}}{T}\right) \text{ with } T \in \mathbb{R} \qquad 1\text{-}1$$

According to the paper, T is called temperature. This parameter is tuned by minimizing Cross Entropy on the test dataset. No more information is given about this approach except that temperature soften the softmax function. Indeed, when looking at the evolution of softmax for different values of T, one can notice that the slope is softer when temperature increase:

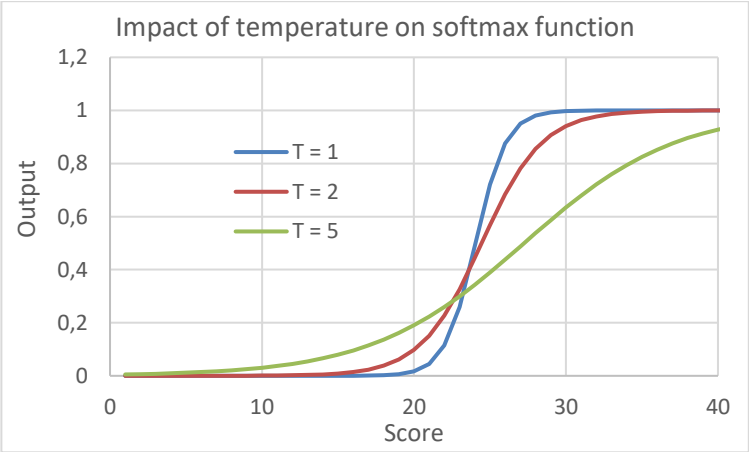


Figure 1 : Shape of Softmax for different value of T

Temperature scaling has been tested with different configuration, and conclusion are common for each experiment. Below are results for AlexNet on Cifar100 and more experiments are presented in Appendix D:

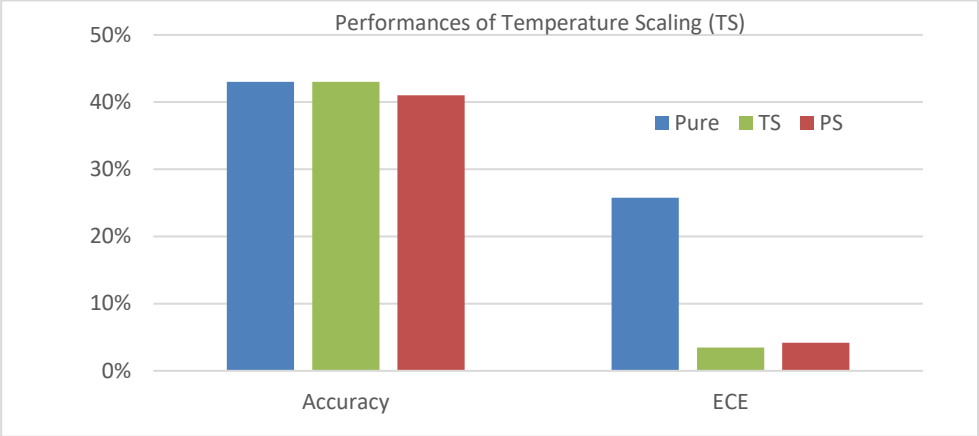


Figure 2 : Metrics for temperature scaling

This technic solves the problem of accuracy drop: dividing scores by a constant will not change the maximum of softmax so the accuracy is conserved. It also decreases ECE better than PS regularization and requires less computation time. Yet, some questions remain:

- How temperature scaling calibrates a network?
- Why Cross Entropy is used as loss function to set T?
- Why TS can't be combined with PS regularization (see Appendix D)?
- Is TS robust again modification of the distribution of inputs?

Indeed, combining TS and PS imply a raise of ECE (from 0,04 to 0,06). To answer to these points, we will extend the theory around temperature scaling and propose an improvement for the way temperature is tuned.

To test Temperature Scaling against inputs dragged from a slightly different distribution, a custom dataset has been created from pictures proposed on the ImageNet dataset [2]. Pictures are chosen to belong to classes already existent in CIFAR-10. Then, the dataset has been tested on pre-trained networks (already trained on CIFAR-10). ECE has been computed on this dataset using pure ensemble and temperature scaling (T tuned with CIFAR-10 validation dataset):

		VGGNet		AlexNet		5xAlexNet Ensemble	
		Accuracy	ECE	Accuracy	ECE	Accuracy	ECE
CIFAR10	Single	78%	17,20%	74%	16,10%	78%	11,93%
	TS	**78%**	**2,70%**	**74%**	**1,30%**	**78%**	**1,39%**
Custom Dataset	Single	67%	26,47%	65%	23,69%	68%	19,51%
	TS	**67%**	**6,06%**	**65%**	**5,74%**	**68%**	**5,71%**

Figure 3 : Accuracy and ECE for networks trained on CIFAR10. Metrics has been computed on a custom dataset from ImageNet database

ImageNet pictures are harder to classify than CIFAR-10. In ImageNet, the concept to detect is not necessarily centered in the picture or big enough. This may explain the drop of accuracy when using our custom dataset. However, this 10% decrease is acceptable, and the table suggest that calibration properties are kept when using a different distribution. We assumed that the differences in ECE value came from the same phenomenon that decrease accuracy.

The second assumption against post-processing technics was that the use of validation test to set temperature parameter may distort the precision of the metrics. To verify this hypothesis, the validation

dataset has been split in two equal part. The first part was used to tune the temperature and the second part to assess the performances of the networks. Below are the results for CIFAR-10 dataset:

		VGGNet		AlexNet		AlexNet Ensemble		ResNet	
		Accuracy	ECE	Accuracy	ECE	Accuracy	ECE	Accuracy	ECE
CIFAR10	Single	78%	16,91%	74%	16,10%	77%	12,71%	81%	15,13%
	TS	**78%**	**2,90%**	**74%**	**1,76%**	**77%**	**2,13%**	**81%**	**1,27%**

Figure 4 : Effect of temperature scaling on different networks trained on CIFAR-10. The validation dataset has been split so that the data used for temperature tuning are different from the data used to compute ECE and Accuracy

Again, ECE seems to be lower with temperature scaling than without. It may suggest that post-processing (specially temperature scaling) do not modify the performances of the network even if they are computed with the validation dataset. According to these results, the entire validation dataset has been used for temperature tuning and metrics assessment.

1.2 Behavior & optimization

As discussed above, overconfidence is taught by Cross Entropy to the network. It learns to output 1 for the correct category and 0 for the others. To do so, it will increase the spreading between scores. Indeed, for a classification problem, if the difference between the highest score and the lowest is important, softmax function will accentuate this gap to return a vector where the highest value is close to 1 and the other values close to 0. In a nutshell, calibration problem is due to an overly important spread of scores.

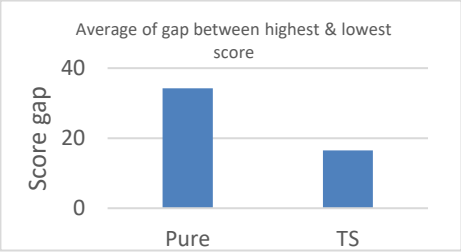


Figure 5 : Reduction of Average gap between the highest and the lowest components of the output vector with and without TS

Temperature scaling simply shrink this spreading. By dividing scores by a constant greater than 1, the gap between the lowest and the highest score will be smaller. This compression will allow softmax function to take a wider range of value in [0,1]. In terms of reliability diagram, that means that inputs are redistributed along x-axis. Accurate classifications from high confidence bins will be moved to lower bins, therefore small peaks appear for the low confidence bins on the reliability diagram:

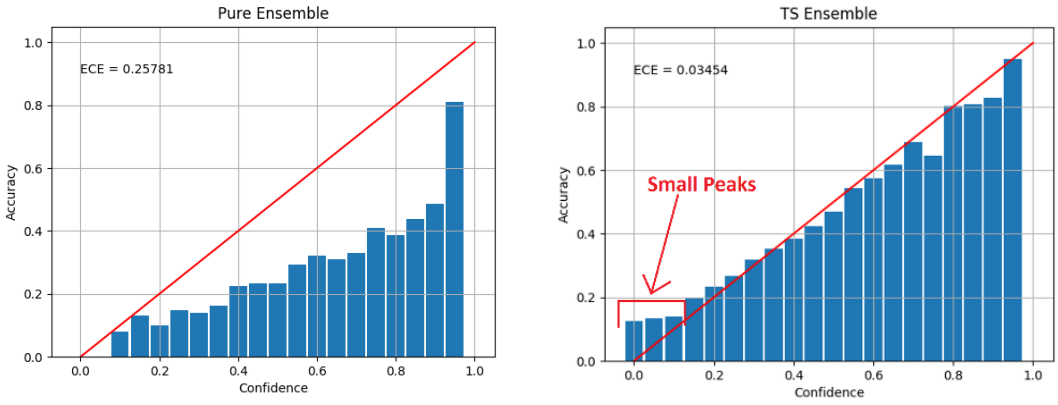


Figure 6 : RD with and without TS

A trade off must be done on the temperature parameter to compress enough scores to improve calibration without reducing the gap too much and increase ECE. The implementation of [1] use Cross Entropy to deal with this trade off. To understand how temperature is tuned, one may look at the loss function:

$$\mathcal{L} = -\sum_{n=1}^{N} \ln\left(softmax\left(\frac{\boldsymbol{h}}{T}\right)_{t^n}\right) \quad 1\text{-}2$$

This loss is minimized by gradient descent. The optimization process is over when the gradient is near to zero :

$$\frac{\partial \mathcal{L}}{\partial T} = 0 \quad 1\text{-}3$$

Now:

$$\frac{d\mathcal{L}}{dT} = \frac{1}{T^2}.\left(\sum_{n=1}^{N} h_{t^n} - \sum_{n=1}^{N}\sum_{k=1}^{K} h_i^n.\sigma\left(\frac{h^n}{T}\right)_i\right) \quad 1\text{-}4$$

So, to minimize the loss function $\mathcal{L}$ with regards to T, we should have $\frac{d\mathcal{L}}{dT} = 0$, that is:

$$\sum_{n=1}^{N} h_{t^n}^n = \sum_{n=1}^{N}\sum_{k=1}^{K} h_i^n.\sigma\left(\frac{h^n}{T}\right)_i \quad 1\text{-}5$$

N is the number of examples in the test dataset. If we divide both side of the equation by N, each term can be interpreted as an average:

- $\frac{1}{N}.\sum_{n=1}^{N} h_{t^n}^n$ is the average of target scores
- $\frac{1}{N}.\sum_{n=1}^{N}\sum_{k=1}^{K} h_i^n.\sigma\left(\frac{h^n}{T}\right)_i$ is the average of all scores where each score is weighted by his probability. We denote $p_{i,n} = \sigma(h^n)_i$ the probability before temperature scaling to simplify notations.

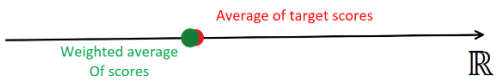
Now, behavior of temperature tuning is studied in different situations. Below are considered neural networks with calibration problem, that is the outputs of these networks are close to one-hot vector (which is synonym of overconfidence). So:

$$p_{i,n} \approx \begin{cases} 1 \; if \; i = \arg\max(\boldsymbol{h}_n) \\ 0 \text{ otherwise} \end{cases}$$

1) Highly Accurate Network: High accuracy may be interpreted as $h_{t^n}^n = \max(h^n)$ almost every time. Furthermore, this network is highly confident so $p_{i,n} = 1 \text{ for } i = \arg\max(h^n)$ and 0 elsewhere. So:

$$\sum_{n=1}^{N}\sum_{k=1}^{K} h_i^n . p_{i,n} \approx \sum_{n=1}^{N} \max(h^n) \approx \sum_{n=1}^{N} h_{t^n}^n \qquad 1\text{-}6$$

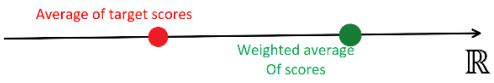
The gradient condition is already respected so temperature scaling is not necessary to satisfy the minimization of loss function. Yet, highly accurate networks rarely have calibration issue, so the deployment of temperature scaling is not relevant in this situation.



2) Poorly accurate Network: Considering a poorly accurate network with strong calibration issue. In this situation, the assumption that $h_{t^n}^n \approx \max(h^n)$ almost every time is not relevant anymore:

$$\begin{aligned} & h_{t^n}^n \neq \max(h^n) \text{ most of the time} \qquad (not\ accurate) \\ & \sum_{n=1}^{N}\sum_{k=1}^{K} h_i^n . p_{i,n} \approx \sum_{n=1}^{N} \max(h^n) \qquad (overconfidence) \end{aligned} \qquad 1\text{-}7$$

In this case, average of **target** score is lower than the **weighted average** of scores. So, we must strongly modify temperature to decrease the weighted average:



Minimizing Cross Entropy with regards to T results in an important contraction of score, which improve calibration as discussed above. So, temperature scaling is efficient to decrease ECE in this case.

3) Intermediate accurate Network: We consider a network with accuracy around 70%-80% with calibration issue. In this situation:

$$h_{t^n}^n = \max(h^n) \text{ often}$$
$$\sum_{n=1}^{N}\sum_{k=1}^{K} h_i^n . p_{i,n} \approx \sum_{n=1}^{N} \max(h^n) \qquad 1\text{-}8$$

Average target scores and weighted average of scores are no very distant, so temperature scaling will have little effect on calibration.

Thereby, the value of T will be close to 1, what is equivalent to a small contraction of scores. Our hypothesis is that it exists a better value of T that will have greater effect on ECE (and so on calibration).

Indeed, relative distance between ECE with and without temperature scaling ($\frac{ECE(Single)-ECE(TS)}{ECE(Single)}$) has been computed for different dataset with different accuracy.

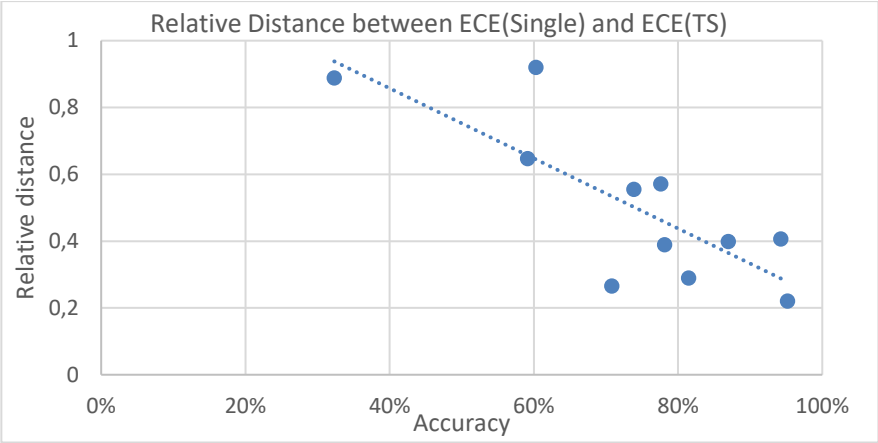


Figure 7 : Relative improvement of ECE for different Accuracy. Each point represents how much the ECE was improved by temperature scaling on a network with the associated accuracy

This graph represents the amount of improvement bring by temperature scaling for different networks with different accuracy. The higher the point is, the better TS performs. It demonstrates that Temperature scaling is less efficient on networks with higher accuracy (points are lower when accuracy is better). To confirm our theory, a new loss function is proposed to tune the temperature with regards to calibration. The objective is to obtain a value of temperature parameter that minimize ECE regardless of the accuracy of the network.

1.3 Temperature scaling improvement

Neural networks have mostly overconfidence problem rather than under-confidence problem. Thereby, reliability diagram is often under the identity line, and confidence is superior to accuracy on each bin. Then, one can reformulate ECE with temperature scaling:

$$ECE = \sum_{k=1}^{N} \frac{Card(B_k)}{n} |Acc(B_i,T) - Conf(B_i,T)| \qquad 1\text{-}9$$

Now, accuracy is independent from T, so $Acc(B_i,T) = Acc(B_i)$. Furthermore, with the assumption $Acc(B_i) \leq Conf(B_i), \forall i$:

$$ECE = -\sum_{k=1}^{N} \frac{Card(B_k)}{n}.Acc(B_i) + \sum_{k=1}^{N} \frac{Card(B_k)}{n}.Conf(B_i,T) \qquad 1\text{-}10$$

Then,

$$ECE \approx |\overline{Acc} - \overline{Conf}(T)| \qquad 1\text{-}11$$

Absolute value can be substituted by a square function that it becomes differentiable. Consequently, the following loss function is suggested:

$$\mathcal{L} = \left(\overline{Acc} - \overline{Conf}(T)\right)^2 \quad \text{1-12}$$

This loss function is referred as *modesty loss*. It has two major advantages. Firstly, it is directly drag from the definition of ECE which make it more adapted to solve calibration problem. Secondly, its behavior is independent from the network and the dataset.

Cross Entropy have different behavior regarding the type of network, the accuracy, the confidence, the dataset, etc. Consequently, it is difficult to propose a minimization strategy that will perform well on every situation. The implementation provided by [1] used a general optimizer sacrificing performances to improve generalization. With modesty loss, one can design a generalized minimization approach regardless of the configuration used.

Indeed, the bar graph below shows the gap between the average of target scores and the weighted average of scores. Cross Entropy is minimum if this gap is equal to zero. "*Pure*" represent the gap without Temperature Scaling. The "Generic Optimizer" is the optimizer as implemented by [1] and the "Adapted Optimizer" was set differently for each network to minimize correctly the Cross Entropy:

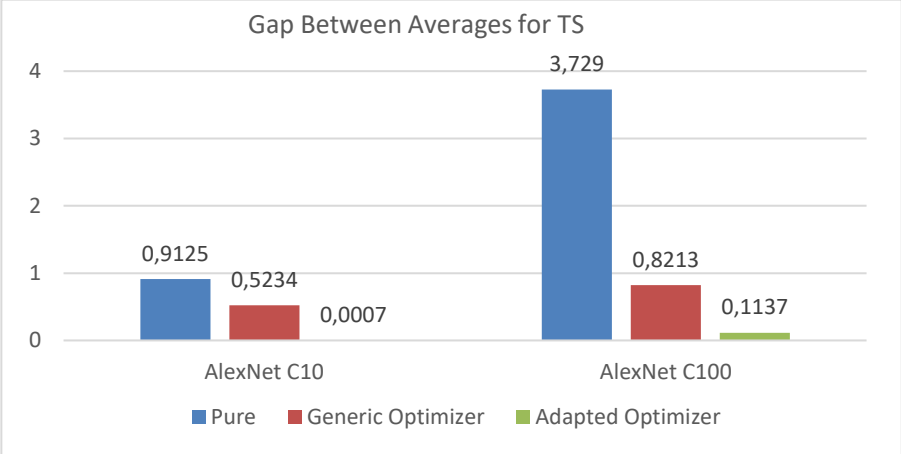


Figure 8 : Gap between the averages for Temperature scaling

Well-tuned optimizer allows temperature scaling to reach lower value of Cross Entropy. However, the objective is to create a framework that can be easily adapted to every classifier. To do so, the optimizer must be capable of minimize Cross Entropy regardless of the network or the dataset. According to this demand, the implementation suggested by [1] was used for further experiments. Obviously, the trainer for Modesty Loss remains also the same for each experiment.

1.4 Results and comparison

We have seen that temperature scaling surpasses the performances of PS regularization, solving accuracy dropping and improving ECE. Nonetheless, weakness have been spotted regarding the tuning process of temperature. To compare the improvement of our proposition, Modesty loss is assessed on different configuration (several networks architecture and different datasets). During the test phase, the optimization process remains the same for each configuration.

		Single		TS Cross Entropy		TS Modesty Loss	
		Accuracy	ECE	Accuracy	ECE	Accuracy	ECE
AlexNet	CIFAR 10	74%	16,1%	74%	7,2%	**74%**	**1,3%**
	CIFAR 100	32%	36,4%	32%	4,0%	**32%**	**3,9%**
	ImageNet	57%	2,1%	57%	4,9%	**57%**	**1,8%**
	SVHN	87%	9,5%	87%	5,7%	**87%**	**4,4%**
VGGNet	CIFAR 10	78%	17,2%	78%	10,5%	**78%**	**2,7%**
	CIFAR 100	60%	15,6%	**60%**	**1,3%**	60%	1,3%
	ImageNet	71%	2,8%	71%	2,0%	**71%**	**2,0%**
	SVHN	94%	4,4%	94%	2,6%	**94%**	**2,2%**
ResNet	CIFAR 10	81%	14,9%	81%	10,6%	**81%**	**0,7%**
	CIFAR 100	59%	29,4%	59%	10,4%	**59%**	**1,4%**
	ImageNet	78%	5,0%	78%	2,1%	**78%**	**1,8%**
	SVHN	95%	3,8%	95%	3,0%	**95%**	**0,6%**

Figure 9 : ECE and Accuracy for different configuration. Metrics have been computed for Single Network, TS with Cross Entropy and TS with Modesty Loss

Modesty Loss perform better than Cross Entropy for almost every configuration. One can notice that TS Cross Entropy may increase ECE for already calibrated networks (such as AlexNet on ImageNet). This issue is solved by Modesty Loss.

We decided to reduce computation time by reducing the number of epochs during the training process. However, temperature scaling has not impact on accuracy, so it can be applied on state-of-the-art methods directly for working with high accuracy approaches. Our results are computed with less accurate networks without loss of generality.

Appendix D provides reliability diagrams for various configuration. Modesty Loss solve the decrease of efficiency issue bring by temperature scaling with Cross Entropy:

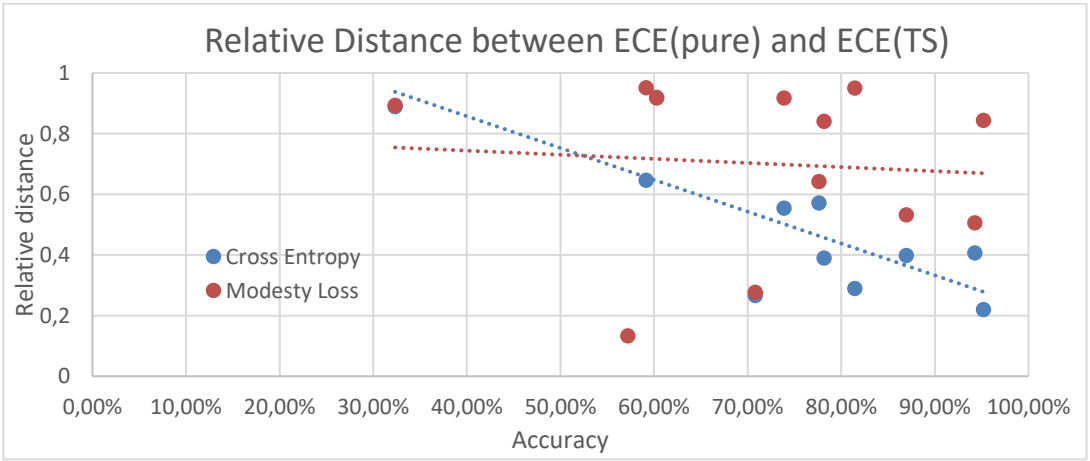


Figure 10 : Relative distance of ECE for different value of accuracy

In average, the relative distance between ECE(Pure) and ECE (TS Modesty Loss) is independent from the accuracy of the network. This allows us to perform well on more accurate networks. Indeed, the red dots represents how much TS with Modesty Loss improve calibration. The slope of the red line (which is the linear approximation of the red dots) is lower than the slope of the blue line (which is the linear approximation of the blue dots computed with TS and Cross Entropy loss). This testify that the efficiency of TS is less affected by accuracy with Modesty Loss than with Cross Entropy.

In conclusion, Modesty Loss manage to resolve the issue caused by Cross Entropy minimization. This loss is more adapted to calibration problem and allow networks to reach very low ECE. Temperature scaling is a simple, efficient et quick method to calibrate a multi-classifier, it can easily be implemented and train as a module on already trained networks. A Pytorch implementation is available at: https://github.com/SteevenJ7/Temperature-Scaling-Modesty-Loss.

2 Bibliographie

[1] G. Chuan, P. Geoff, S. Yu et W. Kilian Q., «On Calibration of Modern Neural Networks,» 2017.

[2] S. V. Lab, «www.image-net.org,» Stanford University, Princeton University , 2016. [En ligne].