Supplementary Material: Calibrating Deep Neural Networks using Explicit Regularisation and Dynamic Data Pruning

Rishabh Patra[1§] Ramya Hebbalaguppe[2§] Tirtharaj Dash[1] Gautam Shroff[2] Lovekesh Vig[2]
[1] APPCAIR, BITS Pilani, Goa Campus [2] TCS Research, New Delhi

October 20, 2022

In this supplementary document, We provide an detailed training algorithm corresponding to Procedure 2 in the main text. Further, we provide some results of calibration for the ResNet32 model in addition to the results in the main text. Our datasets, codes and the resulting models, of all our experiments (shown in the main paper and in the supplementary), will be made available publicly after acceptance.

1 Datasets

We validate our proposed approach on benchmark datasets for image classification. We chose CIFAR-10/100 datasets [1], MendelyV2 (Medical image classification [5]), SVHN[2], and Tiny ImageNet[3]. In all our experiments, we calibrate ResNet-50 [2] and measure the calibration performances using our proposed calibration technique and several other existing techniques. For all experiments, the train set is split into 2 mutually exclusive sets: (a) training set containing 90% of samples and (b) validation set: 10% of the samples. The same validation set is used for post-hoc calibration.

2 A detailed pruning based learning procedure

We provide additionally a detailed version of Procedure 2 from the main text. Procedure 1 details each step of our pruning based learning procedure, as used to obtain calibration with reduced training times, translating to reduced carbon footprint and easier recalibration.

3 Additional Results with ResNet32

3.1 Experiments Parameters

The experimental parameters remain largely the same as in our ResNet50 experiments. For CIFAR10, we train the models for a total of 160 epochs using an initial learning rate of 0.1. The learning rate is reduced by a factor of 10 at the 80^{th} and 120^{th} epochs. The DNN was optimized using Stochastic Gradient Descent (SGD) with momentum 0.9 and weight decay set at 0.0005. Further, the images belonging to the trainset are augmented using random center cropping, and horizontal flips. For CIFAR100, the models are trained for a total of 200 epochs with a learning rate of 0.1, reduced by a factor of 10 at the 100^{th} and 150^{th} epochs. The batch size is set to 1024. Other parameters for training on CIFAR100 are the same ones used for CIFAR10. For Tiny-Imagenet, we follow the same training procedure used by [8]. The models are trained for 100 epochs, with a batch size of 1024. For the Huber loss hyperparameter α, we perform a grid search over values $\{0.001, 0.005, 0.01, 0.05, 0.1\}$. The setting $\alpha = 0.005$ gave the best calibration results across all datasets, and hence we use the same value for all the experiments. For the regularization parameter λ, we perform a grid search over $\{0.1, 0.5, 1, 5, 10, 25, 50\}$, choosing λ with the least ECE and ECE (S_{95}). In our ResNet32 experiments, we find that $\lambda = 25$ gives the least calibration error in both the metrics for CIFAR10. For CIFAR100, $\lambda = 5$ is the optimal parameter for low calibration errors. To set the pruning frequency, we again perform a grid search over $\{5, 10, 25, 50, 100\}$. We find that pruning every 25 epochs is optimal for CIFAR10, whereas pruning every 50 epochs is optimal for CIFAR100. Needless to say, we identify

[§] Equal contribution
[1] https://www.cs.toronto.edu/~kriz/cifar.html
[2] http://ufldl.stanford.edu/housenumbers/
[3] https://image-net.org/

Procedure 1 Our pruning-based learning procedure. The procedure takes as inputs: A dataset of n instances: $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, An untrained neural network $\mathcal{N}$ with structure π and parameters $\boldsymbol{\theta}$, Maximum number of training epochs: $MaxEpochs$, Batch-size: $b \leq n$, Focal loss parameter: γ, Huber Loss parameter: α, Regularization parameter: λ, Learning rate for SGD: η, Weight decay parameter for SGD: β, EMA factor for smoothing: κ, Prune fraction: $\epsilon \in (0, 100)$, A set of pruning epochs during training: $\mathbf{ep}$; and returns: a trained model. The procedure assumes a parameter update procedure BACKPROPWITHSGD.

```
procedure TRAINDNNWITHDATAPRUNING($D$,$\mathcal{N}$,$\pi$,$\boldsymbol{\theta}$,$MaxEpochs$,$b$,$\gamma$,$\alpha$,$\lambda$,$\eta$,$\beta$,$\kappa$,$\epsilon$,$\mathbf{ep}$)
    Let $De = \{(\mathbf{x}_i, y_i, 0)\}_{i=1}^{n}$ where $(\mathbf{x}_i, y_i) \in D\}$, $i \in \{1, \ldots, n\}$
    Number of training batches: $nb = \lceil \frac{n}{b} \rceil$
    Let $B_1, \ldots, B_{nb}$ be the mini-batches of data instances from $De$
    Initialise $\boldsymbol{\theta}$ to small random numbers
    for training epoch $ep$ in $\{1, \ldots, MaxEpochs\}$ do
        for $B_i \in \{B_1, \ldots, B_{nb}\}$ do
            Mean accuracy in batch $i$: $acc = 0$
            Mean confidence in batch $i$: $conf = 0$
            Mean focal loss: $\mathcal{L}_{FL} = 0$
            for each $(\mathbf{x}_k, y_k, e_k) \in B_i$ do
                $\hat{\boldsymbol{y}} = \mathcal{N}(\mathbf{x}_k; (\pi, \boldsymbol{\theta}))$
                $\hat{y} = \arg\max_i \hat{\boldsymbol{y}}$
                $c = \max(\hat{\boldsymbol{y}})$
                $acc = acc + \mathbb{I}(\hat{y} = y_k)$
                $conf = conf + c$
                $\mathcal{L}_{FL} = \mathcal{L}_{FL} + \textsc{ComputeFocalLoss}(\mathrm{onehot(y_k)}, \mathrm{\hat{y}}, \gamma)$
                $e_k = \kappa c + (1 - \kappa) e_k$
            end for
            $acc = acc/b$
            $conf = conf/b$
            $\mathcal{L}_{FL} = \mathcal{L}_{FL}/b$
            Calculate Huber loss: $\mathcal{L}_H = \textsc{ComputeHuberLoss}(acc, conf, \alpha)$
            Calculate total loss: $\mathcal{L}_{total} = \mathcal{L}_{FL} + \lambda \mathcal{L}_H$
            Update parameters of $\mathcal{N}$: $\boldsymbol{\theta} = \textsc{BackPropWithSGD}(\mathcal{L}_{total}, \pi, \boldsymbol{\theta}, \eta, \beta)$
        end for
        Update the instances in $De$ with updated EMA-scores computed above
        if epoch $ep \in \mathbf{ep}$ then
            $De = \textsc{PruneUsingEMA}(De, \epsilon)$
        end if
    end for
end procedure
```

our optimal parameters as those which provide the least calibration error.

Effect of using an EMA score to track evolution of prediction confidences:

We show how the TE, ECE, ECE (S_{95}), $|S_{95}|$ and S_{99} vary with increasing the EMA factor κ. There isn't any noticeable trend of metrics across varying κ. However, using an EMA score to track confidences across epochs, and subsequently using this EMA score to prune training instances is a practically appealing method since it produces

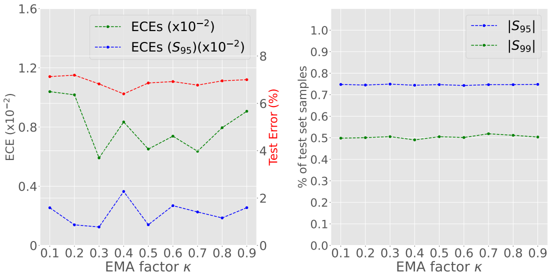

Figure 1: **Effect of varying the EMA factor** (κ) on training a ResNet50 on CIFAR-10 dataset. **Left:** Plots out the ECE, ECE (S_{95}), and TE vs. EMA factor, κ. **Right:** Study of the effect of $|S_{95}|$ and $|S_{99}|$ on varying κ. The best ECE and ECE ($|S_{95}|$) are achieved at $\kappa = 0.3$. Varying κ has no noticeable effect of $|S_{95}|$ and $|S_{99}|$

near perfect calibration of the high confidence instances.

3.2 Experimental Results

Tab. 1 shows the TE, ECE and AUROC for refinement of the ResNet32 models trained on CIFAR10. We notice that MDCA outperforms our proposed approach in terms of ECE. However, it is to be noted that our proposed approach is better calibrated than all the other SOTA approaches. Tab. 2 shows the ECE (S_{95}) and $|S_{95}|$ for our ResNet32 experiments. Here, we obtain the least ECE (S_{95}) against all other SOTA approaches, which is in alignment with our claims of our approach being appealing for practical scenarios. Further, we obtain better $|S_{95}|$ than other Focal loss based methods (ie. MDCA, FLSD).

References

[1] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[3] Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *IEEE/CVF CVPR*, June 2022.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[5] Daniel Kermany, Kang Zhang, Michael Goldbaum, et al. Labeled optical coherence tomography (oct) and chest x-ray images for classification. *Mendeley data*, 2(2), 2018.

[6] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*, pages 2805–2814, 2018.

[7] Gongbo Liang, Yu Zhang, Xiaoqin Wang, and Nathan Jacobs. Improved trainable calibration method for neural networks on medical imaging classification. *CoRR*, abs/2009.04057, 2020.

[8] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. Calibrating deep neural networks using focal loss, 2020.

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

Dataset	**Model**	**BS [1]**			**DCA [7]**			**LS [9]**			**MMCE [6]**			**FLSD [8]**			**FL + MDCA [3]**			**Ours (FLSD+H+P$_{\text{EMA}}$)**		
		ECE	TE	AUROC	ECE	TE	AUROC	ECE	TE	AUROC	ECE	TE	AUROC	ECE	TE	AUROC	ECE	TE	AUROC	ECE	TE	AUROC
CIFAR10	ResNet32	2.83	7.67	90.35	3.58	17.13	87.39	3.06	7.71	85.25	4.16	7.36	90.81	5.27	7.82	91.24	**0.93**	**7.18**	**91.62**	1.10	8.33	91.42
CIFAR100	ResNet32	7.87	36.78	**85.67**	7.82	42.91	82.46	7.44	34.66	83.52	15.09	**33.33**	84.48	3.28	35.7	82.91	**3.06**	35.47	83.39	3.18	35.41	82.36

Table 1: Calibration measure ECE (%) score), Test Error (TE) (%) and AUROC (refinement) in comparison with various competing methods. We use $M = 10$ bins for ECE calculation. We outperform most of the baselines across various popular benchmark datasets, and architectures in terms of calibration, while maintaining a similar accuracy and a similiar refinement (AUROC.)

Dataset	**Model**	**BS [1]**		**DCA [7]**		**LS [4]**		**MMCE [6]**		**FLSD [8]**		**FL + MDCA [3]**		**Ours (FLSD+H+P$_{\text{EMA}}$)**	
		ECE (S_{95})	$\lvert S_{95}\rvert$	ECE (S_{95})	$\lvert S_{95}\rvert$	ECE (S_{95})	$\lvert S_{95}\rvert$	ECE (S_{95})	$\lvert S_{95}\rvert$	ECE (S_{95})	$\lvert S_{95}\rvert$	ECE (S_{95})	$\lvert S_{95}\rvert$	ECE (S_{95})	$\lvert S_{95}\rvert$
CIFAR10	ResNet32	1.49	82.44	1.29	54.74	1.55	74.46	2.54	**87.80**	2.36	40.52	1.56	57.63	**0.002**	72.36
CIFAR100	ResNet32	3.47	33.34	2.26	20.28	3.31	28.40	7.84	**46.63**	0.54	16.47	0.11	17.07	**0.008**	17.37

Table 2: Top-label calibration measure ECE (S_{95}) (% score) and $|S_{95}|$ (percentage of total number of test samples with predictive confidences ≥ 0.95) in comparison with various competing methods. We use $M = 10$ bins for ECE (S_{95}) calculations. We outperform all the baselines across various popular benchmark datasets, and architectures in terms of calibration. While we do not outperform all calibration methods in terms of $|S_{95}|$,it is to be noted that we obtain a higher $|S_{95}|$ than (FLSD, MDCA).