On Calibration of Modern Neural Networks

Chuan Guo, Geoff Pleiss, Yu Sun, Kilian Q. Weinberger

Alex Fedorov

The Georgia State University/Georgia Institute of Technology/Emory University
Center for Translational Research in Neuroimaging and Data Science (TReNDS)

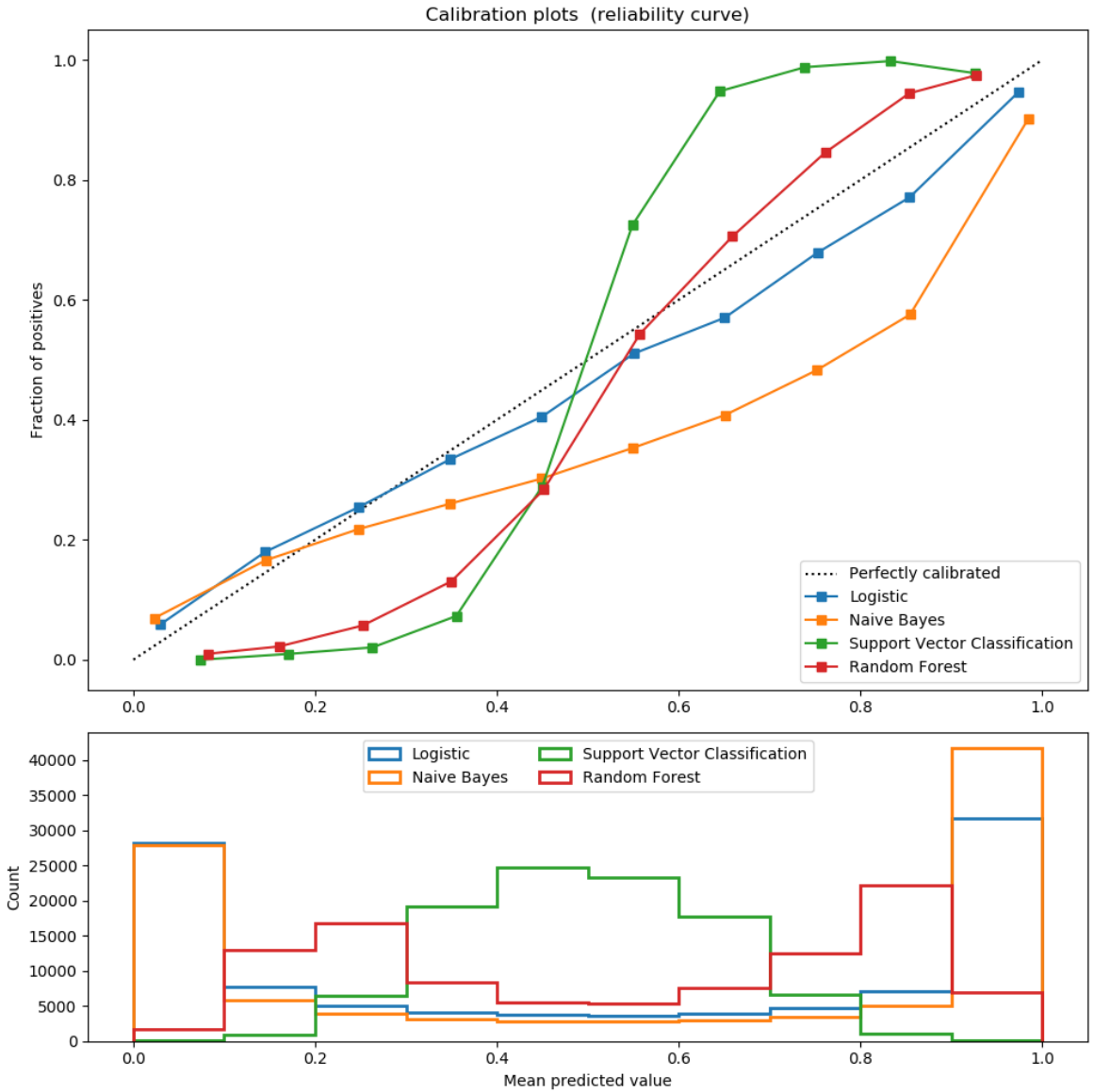
August 30, 2019

Confidence calibration

- **Confidence calibration** - the problem of predicting probability estimates representative of the true correctness likelihood
- **Why?**
  - The probability associated with the predicted class label should reflect its ground truth correctness
    - Important for self-driving cars
    - Diagnosis in automated health care
  - Model interpretability
    - Extra information for trust
    - Can be passed further to other models of the system

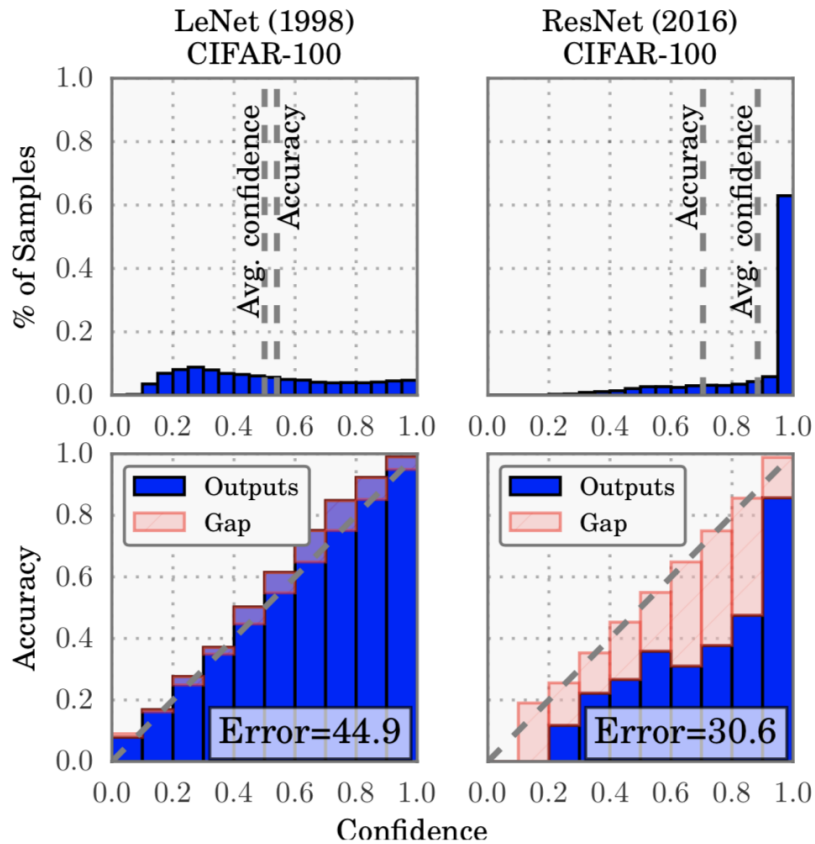
Applicable to Machine Learning

https://scikit-learn.org/stable/modules/calibration.html



Modern vs old

- Modern networks are poorly calibrated unlike old
  - a 5-layer LeNet (LeCun et al., 1998) is 😇
  - a 110-layer ResNet (He et al., 2016) on the CIFAR-100 dataset is ☠️🚩.



Perfect calibration

Supervised multi-class classification:

- The input $X \in \mathcal{X}$ and label $Y \in \mathcal{Y} = 1, ..., K$
- Follows $\sim \pi(X, Y) = \pi(Y|X)\pi(X)$
- The Neural Network — $h(X) = (\hat{Y}, \hat{P})$

The *perfect calibration* is

$$\mathbb{P}(\hat{Y} = Y|\hat{P} = P) = p, \forall p \in [0, 1] \tag{1}$$

Reliability diagrams

- **Reliability Diagrams** are a visual representation of model calibration (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005)
- These diagrams plot expected sample accuracy as a function of confidence
- If the model is perfectly calibrated – i.e. if (1) holds – then the diagram should plot the identity function. Any deviation from a perfect diagonal represents miscalibration.

Expected accuracy and average confidence

- Let B_m be the set of indices $\in I_m = (\frac{m-1}{M}, \frac{m}{M}]$. The expected accuracy of B_m is

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i)$$

- The average confidence within bin B_m is defined as:

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

- $acc(B_m)$ and $conf(B_m)$ approximate the left-hand and right-hand sides of (1) respectively for bin B_m
- A *perfectly calibrated model* will have $acc(B_m) = conf(B_m)$

Expected Calibration Error

- The **Expected Calibration Error** (ECE) is used to summarize calibration as statistics.
- One notion of miscalibration is the difference in expectation between confidence and accuracy

$$\mathbb{E}_{\hat{P}}\Big[\Big|\mathbb{P}(\hat{Y} = Y|\hat{P} = P) - p\Big|\Big] \tag{2}$$

- It is approximates by (Naeini et al., 2015) as:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n}\Big|acc(B_m) - conf(B_m)\Big| \tag{3}$$

Mamixum Calibration Error

- For high-risk application we may wish to minimize the worst-case deviation between confidence and accuracy

$$\max_{p \in [0,1]} \left[\left| \mathbb{P}(\hat{Y} = Y | \hat{P} = P) - p \right| \right] \tag{4}$$

- **The Mamixum Calibration Error** (MCE) is defined as:

$$MCE = \max_{m \in 1,\ldots,M} \left| acc(B_m) - conf(B_m) \right| \tag{5}$$

Negative log likelihood

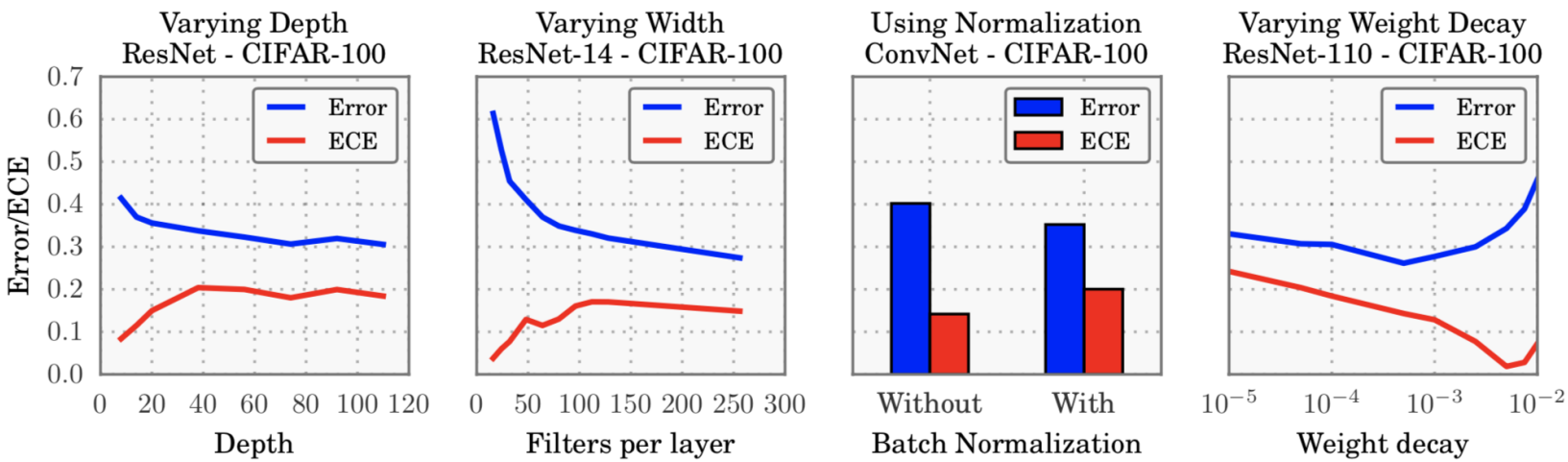
- Negative log likelihood is a standard measure of a probabilistic model's quality (Friedman et al., 2001)
- It is also referred to as the cross entropy loss in the context of deep learning (Bengio et al., 2015)
- Given a probabilistic model $\pi(Y|X)$ and n samples, NLL is defined as:

$$\mathcal{L} = -\sum_{i=1}^{n} log(\hat{\pi}(y_i|\mathbf{x}_i)) \tag{6}$$

- It is a standard result (Friedman et al., 2001) that, in expectation, NLL is minimized if and only if $\hat{\pi}(Y|X)$ recovers the ground truth conditional distribution $\pi(Y|X)$.

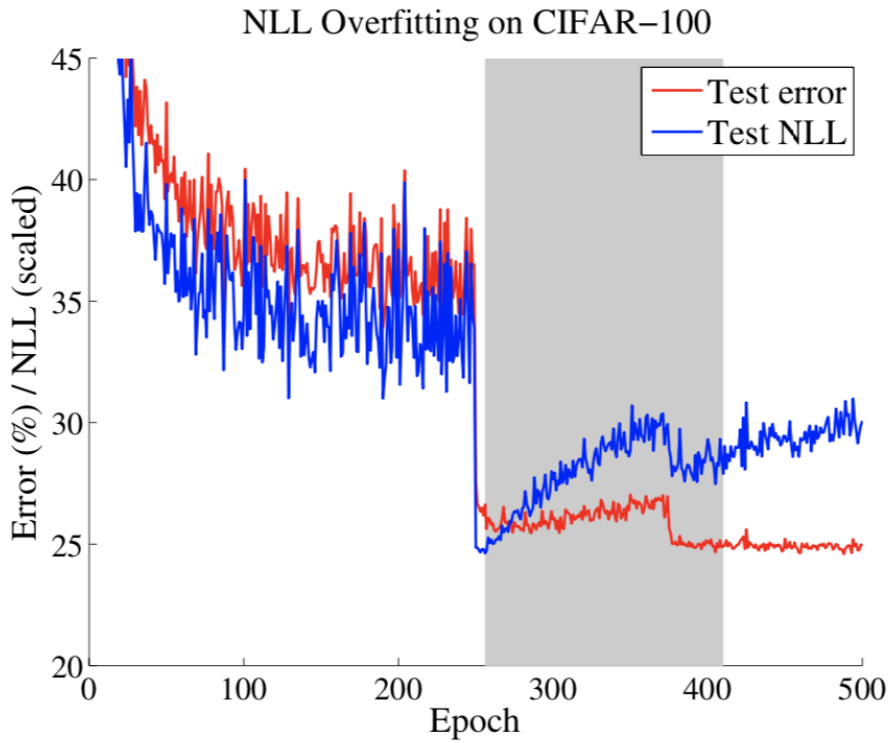
Observing Miscalibration

- *Increasing depth and width* **may reduce classification error — negatively affect model calibration**
- The models trained *with Batch Normalization* **tend to be more miscalibrated**
- The training *with less weight decay* **has a negative impact on calibration**.



NLL and Calibration

- The network learns better classification accuracy at the expense of well-modeled probabilities.
- These high capacity models are not necessarily immune from overfitting, but rather, overfitting manifests in probabilistic error rather than classification error.

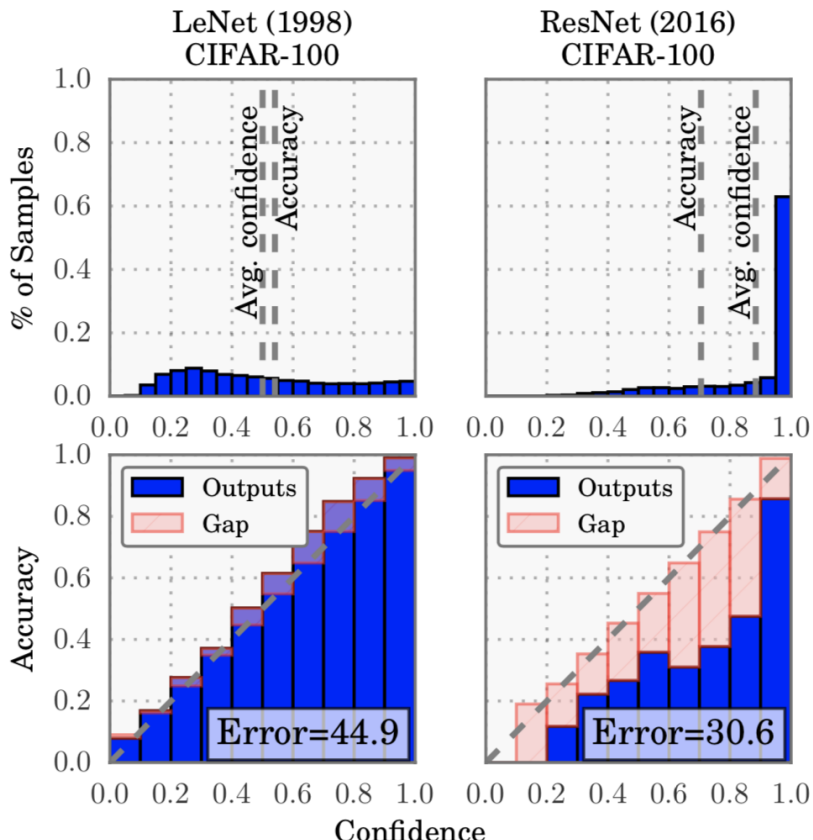


Calibration Methods

- Histogram binning (Zadrozny & Elkan, 2001)
- Isotonic regression (Zadrozny & Elkan, 2002)
- Bayesian Binning into Quantiles (BBQ) (Naeini et al., 2015)
- Platt scaling (Platt et al., 1999, Niculescu-Mizil & Caruana, 2005)

Histogram Binning

- All uncalibrated predictions $\hat{p}^i$ are divided into mutually exclusive bins $B_1, ..., B_M$.
- Each bin is assigned a calibrated score θ_m, i.e. if $\hat{p}_i$ is assigned to bin B_m, then $\hat{q}^i = \theta_m$
- At test time, if prediction $\hat{p}_{te}$ falls into bin B_m, then the calibrated prediction $\hat{q}_{te}$ is θ_m.



Histogram Binning & Isotonic regression

- Histogram binning

$$\min_{\theta_1,\ldots,\theta_M} \sum_{m=1}^{M} \sum_{i=1}^{n} \mathbf{1}(a_m \leq \hat{p}_i < a_{m+1})(\theta_m - y_i)^2$$

- Isotonic Regression

$$\min_{M,\theta_1,\ldots,\theta_M,a_1,\ldots,a_{M+1}} \sum_{m=1}^{M} \sum_{i=1}^{n} \mathbf{1}(a_m \leq \hat{p}_i < a_{m+1})(\theta_m - y_i)^2$$

$$\text{subject to } 0 = a_1 \leq a_2 \leq \cdots \leq a_{M+1} = 1,$$

$$\theta_1 \leq \theta_2 \leq \cdots \leq \theta_M$$

Bayesian Binning into Quantiles

- BBQ marginalizes out all – possible binning schemes to produce $\hat{q}$
- BBQ performs Bayesian averaging of the probabilities produced by each scheme

$$\mathbb{P}(\hat{q}_{te}|\hat{p}_{te}, D) = \sum_{s \in \mathcal{S}} \mathbb{P}(\hat{q}_{te}, S = s|\hat{p}_{te}, D)$$
$$= \sum_{s \in \mathcal{S}} \mathbb{P}(\hat{q}_{te}|\hat{p}_{te}, S = s, D)\mathbb{P}(S = s|D),$$

where $\mathbb{P}(\hat{q}_{te}|\hat{p}_{te}, S = s, D)$ is a the calibrated probability using scheme s

Platt Scaling

- Platt scaling (Niculescu-Mizil & Caruana, 2005), learns scalar parameters $a, b \in \mathbb{R}$ and outputs $\hat{q} = \sigma(az_i + b)$ as the calibrated probability
- a and b is optimized over NLL loss
- The parameters of NN should be fixed

Multiclass case: $K > 2$

$$(\hat{y}_i, \hat{p}_i) = NN(\mathbf{x}_i)$$

$$\hat{y}_i = \underset{k}{\text{argmax}}\, z_i^{(k)}$$

$$\sigma_{SM}(\mathbf{z}_i)^{(k)} = \frac{exp(z_i^{(k)})}{\sum_{j=1}^{K} exp(z_i^{(j)})}$$

$$\hat{p}_i^{(k)} = \max_k \sigma_{SM}(\mathbf{z}_i)^{(k)}$$

Extension for binning methods

- Treating the problem as K one-versus-all problems
- Form a binary calibration problem where the label is $\mathbf{1}(y_i = k)$ and the predicted probability is $\sigma(z)_{SM}^{(k)}$
- Obtain $[\hat{q}_i^{(1)}, ..., \hat{q}_i^{(K)}]$
- Predict $\hat{y}_i' = \text{argmax}[\hat{q}_i^{(1)}, ..., \hat{q}_i^{(K)}]$
- New confidence is $\hat{q}_i' = \frac{max[\hat{q}_i^{(1)}, ..., \hat{q}_i^{(K)}]}{\sum_{j=1}^{L} \hat{q}_i^{(j)}}$

Matrix and Vector Scaling

- Let $\mathbf{z}_i$ be the logits vector produced before the softmax layer for input $\mathbf{x}_i$. Matrix scaling applies a linear transformation $\mathbf{W}\mathbf{z}_i + \mathbf{b}$ to the logits

$$\hat{q}_i = \max_k \sigma_{SM}(\mathbf{W}\mathbf{z}_i + \mathbf{b})^{(k)}$$

$$\hat{y}_i' = \operatorname{argmax}_k (\mathbf{W}\mathbf{z}_i + \mathbf{b})^{(k)}$$

- $\mathbf{W}$ is restricted to be diagonal matrix, because of quadratic grows of parameters with number of classes

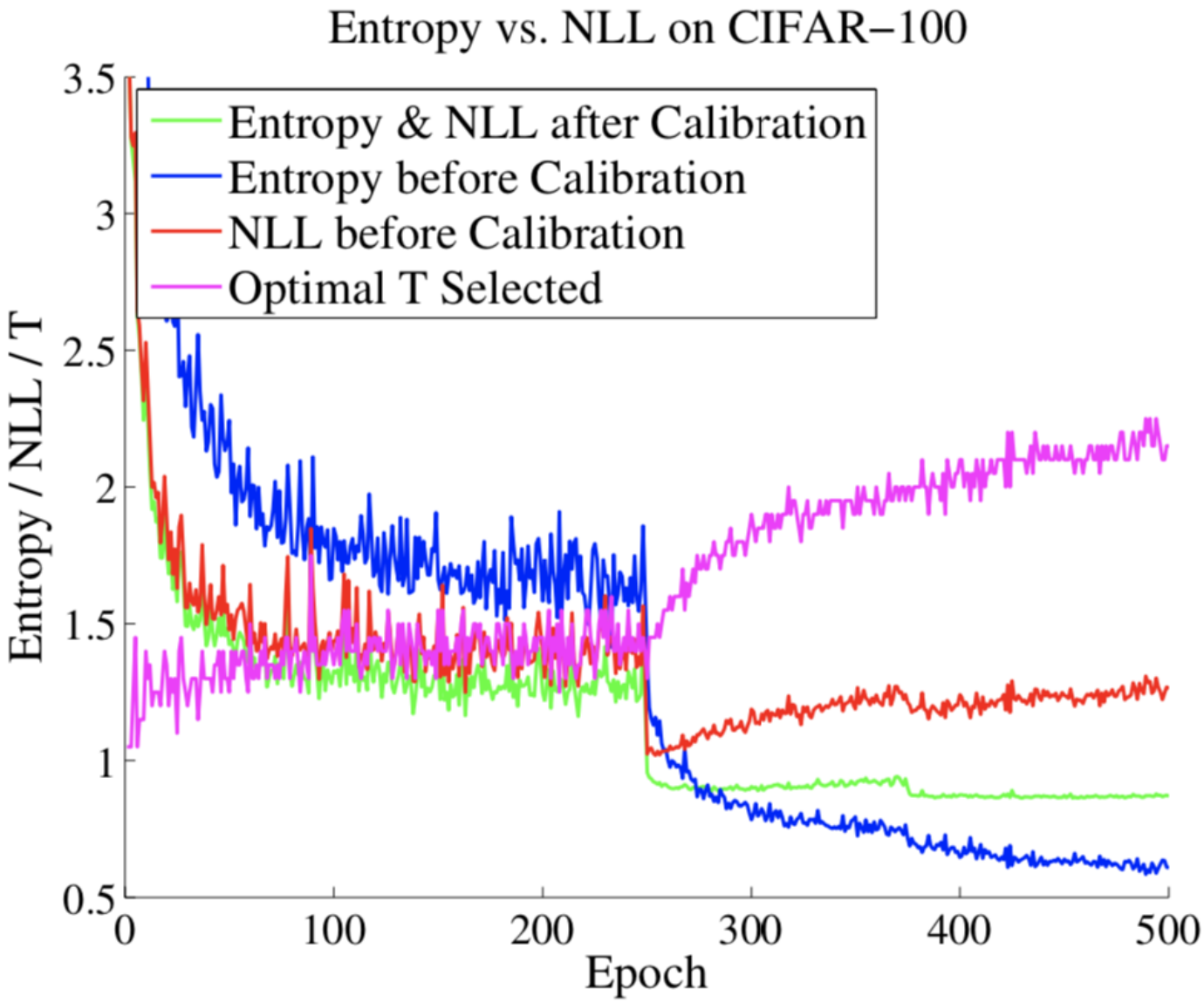
Temperature Scalling

- The simplest extension of Platt scaling, uses a single scalar parameter $T > 0$ for all classes
- Given the logit vector $\mathbf{z}_i$, the new confidence prediction is

$$\hat{q}_i = \max_k \sigma_{SM}\left(\frac{\mathbf{z}_i}{T}\right)^{(k)}$$

- T "softens" the softmax (i.e. raises the output entropy) with $T > 1$.
- As $T \to \inf$, the probability $\hat{q}_i$ approaches $1/K$, which represents maximum uncertainty.
- With $T = 1$, we recover the original probability $\hat{p}_i$.
- As $T \to 0$, the probability collapses to a point mass (i.e. $\hat{q}_i = 1$)
- T is optimized with respect to NLL on the validation set
- Prediction $\hat{y}_i'$ remains unchanged, since T does not change the maximum of the softmax function, temperature scaling does not affect the model's accuracy.

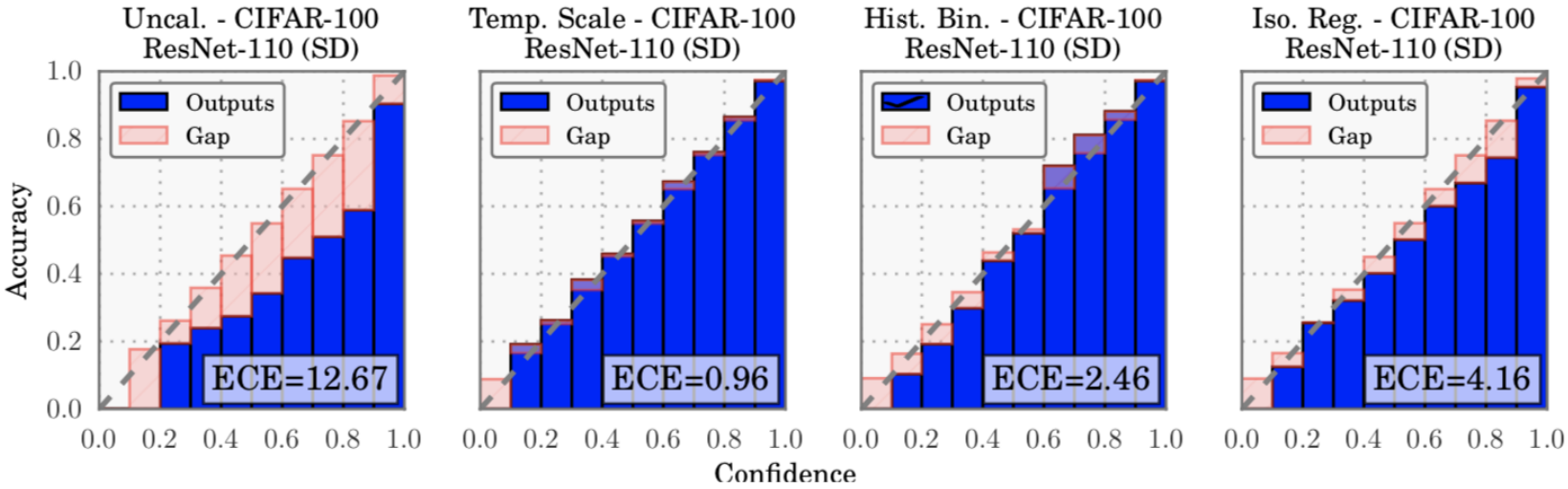
Training on CIFAR100



Results

Dataset	Model	Uncalibrated	Hist. Binning	Isotonic	BBQ	Temp. Scaling	Vector Scaling	Matrix Scaling
Birds	ResNet 50	9.19%	4.34%	5.22%	4.12%	**1.85%**	3.0%	21.13%
Cars	ResNet 50	4.3%	**1.74%**	4.29%	1.84%	2.35%	2.37%	10.5%
CIFAR-10	ResNet 110	4.6%	0.58%	0.81%	**0.54%**	0.83%	0.88%	1.0%
CIFAR-10	ResNet 110 (SD)	4.12%	0.67%	1.11%	0.9%	**0.6%**	0.64%	0.72%
CIFAR-10	Wide ResNet 32	4.52%	0.72%	1.08%	0.74%	**0.54%**	0.6%	0.72%
CIFAR-10	DenseNet 40	3.28%	0.44%	0.61%	0.81%	**0.33%**	0.41%	0.41%
CIFAR-10	LeNet 5	3.02%	1.56%	1.85%	1.59%	**0.93%**	1.15%	1.16%
CIFAR-100	ResNet 110	16.53%	2.66%	4.99%	5.46%	**1.26%**	1.32%	25.49%
CIFAR-100	ResNet 110 (SD)	12.67%	2.46%	4.16%	3.58%	0.96%	**0.9%**	20.09%
CIFAR-100	Wide ResNet 32	15.0%	3.01%	5.85%	5.77%	**2.32%**	2.57%	24.44%
CIFAR-100	DenseNet 40	10.37%	2.68%	4.51%	3.59%	1.18%	**1.09%**	21.87%
CIFAR-100	LeNet 5	4.85%	6.48%	2.35%	3.77%	**2.02%**	2.09%	13.24%
ImageNet	DenseNet 161	6.28%	4.52%	5.18%	3.51%	**1.99%**	2.24%	-
ImageNet	ResNet 152	5.48%	4.36%	4.77%	3.56%	**1.86%**	2.23%	-
SVHN	ResNet 152 (SD)	0.44%	**0.14%**	0.28%	0.22%	0.17%	0.27%	0.17%
20 News	DAN 3	8.02%	**3.6%**	5.52%	4.98%	4.11%	4.61%	9.1%
Reuters	DAN 3	0.85%	1.75%	1.15%	0.97%	0.91%	**0.66%**	1.58%
SST Binary	TreeLSTM	6.63%	1.93%	**1.65%**	2.27%	1.84%	1.84%	1.84%
SST Fine Grained	TreeLSTM	6.71%	2.09%	**1.65%**	2.61%	2.56%	2.98%	2.39%

Results


Uncal. - CIFAR-100
ResNet-110 (SD)
Temp. Scale - CIFAR-100
ResNet-110 (SD)
Hist. Bin. - CIFAR-100
ResNet-110 (SD)
Iso. Reg. - CIFAR-100
ResNet-110 (SD)
Outputs
Gap
ECE=12.67
ECE=0.96
ECE=2.46
ECE=4.16
Accuracy
Confidence
0.0
0.2
0.4
0.6
0.8
1.0

Other directions

- Temperature scaling is commonly used in settings such as knowledge distillation (Hinton et al., 2015) and statistical mechanics (Jaynes, 1957).
- Kuleshov Ermon (2016) **in the online setting**, where the inputs can come from a **adversarial source**.
- Kuleshov Liang (2015) calibrated probabilities when the output space is a **structured object**.
- Lakshminarayanan et al. (2016) use *ensembles of networks* to obtain **uncertainty** estimates.
- Pereyra et al. (2017) **penalize overconfident predictions** as a form of *regularization*.
- Hendrycks Gimpel (2017) use *confidence scores* to determine if samples are **out-of-distribution**

Other directions

- **Bayesian neural networks** (Denker Lecun, 1990; MacKay, 1992)
- Gal Ghahramani (2016) draw a connection between *Dropout* (Srivastava et al., 2014) and *model uncertainty*
- Kendall Gal (2017) outputs a predictive mean and variance for each data point.
- **Deep kernel learning** (Wilson et al., 2016a;b; Al-Shedivat et al., 2016) combines deep neural networks with *Gaussian processes* on classification and regression problems.

Thank you!
Code is available ☺!
https://github.com/gpleiss/temperature$_s$*caling*