Mixup training

Analysis of the impact on calibration of deep neural networks

Moritz Neuhaus, Mikhail Kazantsev, Lomàn Vezin

École Polytechnique

Table of contents

Introduction

Motivation

Although modern DNN show an increase in accuracy they face major calibration issues.

Mixup is a recently proposed method to reduce overconfidence in image classification and decision making DNN, improving overall calibration.

The method is both effective and simple to implement.[2]

New samples are generated during training as a convex combination of random pairs of images and their labels.

Methods used - Data set

Mixup is based on *Vicinal Risk Minimization* principle.

From randomly selected images x_i, x_j and labels y_i, y_j we generate vicinial points $\tilde{x}, \tilde{y}$ as

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j$$

We then train the network on these vicinial points.

Methods used - Loss function

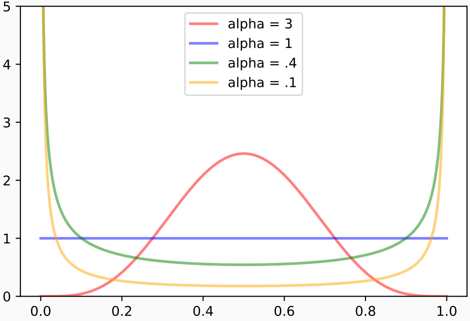
For training the network on the mixup data set we used a new mixup criterion, inspired by [2], using the *Cross Entropy Loss* $\mathcal{L}$ and the same notations

$$\tilde{\mathcal{L}}(\tilde{x}, \tilde{y}) = \lambda \mathcal{L}(\tilde{x}, y_i) + (1 - \lambda) \mathcal{L}(\tilde{x}, y_j).$$

Methods used - λ parameter

The interpolating factor $\lambda \in [0, 1]$ is drawn from the symmetric beta distribution with chosen parameter α. We used values of α from 0.1 to 0.4, proven to show best results.[3]

λ determines the mixing ratio, $\lambda = 0, 1$ corresponds to the original minimisation.



Calibration Metric

For calibration we used the *Expected Calibration Error* metric as proposed in [1][2]. It can be written as

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

where B_m are equally spaced bins for the predictions over n samples.

Implementation on CIFAR-10

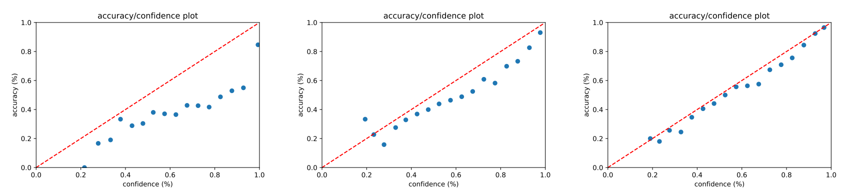
Implementation

For our implementation we first used a wide network `wide_resnet50_2` and then moved on to `resnet18`, yielding better results.

We used the *SGD* optimiser with learning rate 0.003 and momentum 0.9. The criterion used is the *Cross Entropy Loss*.

The model achieved $\sim 70\%$ accuracy on all different instances. We noticed a small difference, roughly less than 2%, between no mixup and the most efficient mixup ($\alpha = 0.4$).

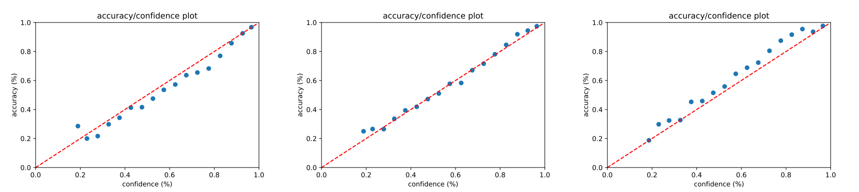
Results



$\alpha = 0,\ ECE = 0.19559$ $\alpha = 0.1,\ ECE = 0.08888$ $\alpha = 0.2,\ ECE = 0.03203$

Figure 1: Accuracy/confidence plot for several α values

Results



$\alpha = 0.3,\ ECE = 0.03127$ $\alpha = 0.4,\ ECE = 0.01761$ $\alpha = 0.7,\ ECE = 0.06188$

Figure 2: Accuracy/confidence plot for several α values

Conclusion

Questions?

References

- Chuan Guo et al. *On Calibration of Modern Neural Networks.* 2017. arXiv: *1706.04599 [cs.LG]*.
- Sunil Thulasidasan et al. *On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks.* 2020. arXiv: *1905.11001 [stat.ML]*.
- Hongyi Zhang et al. "mixup: Beyond Empirical Risk Minimization". In: *CoRR* abs/1710.09412 (2017). arXiv: *1710.09412*. URL: *http://arxiv.org/abs/1710.09412*.