Regularizing Prediction Entropy Enhances Deep Learning with Limited Data

Abhimanyu Dubey
MIT
dubeya@mit.edu

Otkrist Gupta
MIT
otkrist@mit.edu

Ramesh Raskar
MIT
raskar@mit.edu

Iyad Rahwan
MIT
irahwan@mit.edu

Nikhil Naik
Harvard University
naik@fas.harvard.edu

Abstract

Many supervised learning problems require learning with small amounts of training data, since constructing large training datasets could be impractical due to cost, labor, or unavailability of data. For such tasks, constructing deep learning approaches that generalize to new data is difficult. In this paper, we demonstrate the effectiveness of using entropy as a regularizer on image classification tasks involving very small amounts of data. Optimizing with entropy regularization enables neural networks to learn more generalizable feature representations in the penultimate layers. We conduct experiments on training from scratch on limited subsets of CIFAR10 and CIFAR100 as well as on fine-tuning from existing models on three datasets for fine-grained visual recognition (FGVC) and observe significant improvements in classification performance on both tasks.

1 Introduction

A plethora of machine learning problems across scientific domains involve very small amounts of training data. Applications of deep learning in medical imaging [15], fine-grained recognition [16, 14] and domain adaptation [3] have training data that is orders of magnitude lower than traditional image classification datasets such as ImageNet [5] or Places365 [27] which have thousands of training samples for every output class.

The difficulty in obtaining annotated training samples for such tasks cannot necessarily be mitigated easily; in applications such as medical image classification and fine-grained recognition, obtaining annotations is expensive and requires domain experts [14]. In addition, due to concerns around privacy (in medical imaging) or the inability to photograph certain fine-grained categories, obtaining unlabeled raw samples in large amounts is in itself impractical.

The effectiveness of large-scale deep convolutional neural networks across tasks in computer vision [13, 23] have made deep learning the *de facto* choice for image classification. Apart from their high computational complexity, another issue of concern is the requirement of high amounts of training data for generalizable performance. Despite the incredible success of large CNNs trained on ImageNet as generic image feature representations [6, 22], these networks do not naively generalize well to tasks involving limited amounts of training data.

A common approach in such classification problems is to initialize a model on weights obtained by training on a large corpus such as ImageNet, and *fine-tuning* the model on the target task. This approach is preferred for such problems, with extensive analysis on techniques to improve fine-tuning performance [4, 25]. Domain-specific neural network architectures have been designed for some

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

of these problems, such as Bilinear Pooling for fine-grained visual recognition [14], and D-CAN for histology image segmentation [2]. Pereyra *et al.* [19] experimented with the idea of penalizing deep neural network classifiers for confident predictions. While they demonstrate improvements in performance across several tasks, these improvements—especially on image classification—are negligible (0.1% average improvements, which is within the standard deviation of the accuracy across trials).

Penalizing overconfidence, is an interesting proposition, but applicable more strongly when limited training data is present. In the absence of densely sampled data points, the training data is more prone to have sampling biases and may not be representative of the underlying distribution. Hence, it is reasonable to prevent the neural network classifier from making overconfident predictions in this case.

In this paper, we establish the effectiveness of regularizing the entropy of the output predictions (a measure for classifier overconfidence) on image classification problems with limited amounts of training data. We observe that as training data increases, the benefits obtained from this penalty diminish, in accordance with our results. Contrary to Pereyra *et al.* [19], our demonstrated improvements are larger, substantiating the applicability of entropy-based regularization in data-constrained classification tasks.

2 Method

2.1 Motivation

The motivation behind the formulation for entropy regularization stems from penalizing the peakiness of the output probability distribution, that is, we require the conditional probability distribution $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ produced as output by the network for an input sample $\boldsymbol{x}$ (under model with parameters θ) to have mass distributed across the alphabet of Y, that is, across several classes. Peaky distributions have lower entropy [8], and it is evident that we obtain maximum entropy when all events are equally likely, i.e. there is no "surprise" in the observation.

Since fine-tuning a model trained on a large dataset on the target dataset is common practice in image classification, the model used for prediction usually has a much higher capacity than the small target dataset it is being fine-tuned on. This practice hence makes it very easy to overfit to the training data. One method of performing regularization for this problem is early-stopping [20], which involves ceasing training once validation performance begins decreasing. While effective, early-stopping is often plagued by the issue deciding when the model has begun to overfit, and ameliorating that requires keeping subsequent copies of the model, which can be memory intensive.

Additionally, if the available labels are corrupted by labeling noise, we cannot allow training until convergence. For applications where the target classes have high similarity (e.g., fine-grained recognition and classification), training with naive cross-entropy would imply that for each sample, there is no noise present in its label, and only the specified label is present with certainty, consequently leading to overfitting when training.

2.2 Formulation

As we specified earlier, a metric to regularize the confidence and increase confusion in output distributions would increase the entropy of the output conditional probability distribution. Entropy $H(p_\theta(\boldsymbol{y}|\boldsymbol{x}))$ for the conditional probability distribution $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ can be given by:

$$H(p_\theta(\boldsymbol{y}|\boldsymbol{x})) = -\sum_{i=1}^{N} p_\theta(\boldsymbol{y}_i|\boldsymbol{x}) \cdot \log(p_\theta(\boldsymbol{y}_i|\boldsymbol{x})) \tag{1}$$

We can maximize entropy by specifying our learning objective function $\mathcal{L}$ as:

$$\mathcal{L} = \mathcal{L}_{ce}(p_\theta(\boldsymbol{y}|\boldsymbol{x})) - \alpha \cdot H(p_\theta(\boldsymbol{y}|\boldsymbol{x})) \tag{2}$$

Where $\mathcal{L}_{ce}$ denotes the prevalent cross-entropy loss. The new functional $\mathcal{L}$ promotes learning a classifier with a maximum entropy output distribution (under suitable regularization parameter α), while maintaining classification accuracy.

Another interpretation of the same objective function can be drawn from measuring the divergence of the conditional probability distribution from the uniform distribution. One frequently used measure for estimating this metric, that can also be employed in the training of neural networks with cross-entropy, is the Kullback-Leibler(KL) divergence [10]. The asymmetry of this metric gives rise to two forms of regularization. If we write the KL-divergence of $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ from the uniform distribution with mean $\frac{1}{N}$, denoted as $U(\frac{1}{N})$, we get:

$$\mathbb{D}_{\mathsf{KL}}(p_\theta(\boldsymbol{y}|\boldsymbol{x}) \parallel U(\frac{1}{N})) = \sum_{i=1}^{N} p_\theta(\boldsymbol{y}_i|\boldsymbol{x}) \cdot \log\Big(\frac{p_\theta(\boldsymbol{y}_i|\boldsymbol{x})}{N^{-1}}\Big) \tag{3}$$

$$= \log N + \sum_{i=1}^{N} p_\theta(\boldsymbol{y}_i|\boldsymbol{x}) \cdot \log p_\theta(\boldsymbol{y}_i|\boldsymbol{x}) \tag{4}$$

$$= \log N - H(p_\theta(\boldsymbol{y}|\boldsymbol{x})) \tag{5}$$

Hence, we can see that maximizing the entropy H is identical to minimizing one direction of KL-divergence of the output conditional probability distribution from the uniform distribution. Reversing the direction of the divergence yields the label-smoothing regularization (LSR) [24], a technique which involves altering the one-hot label vector to a smoother version, by replacing the mass at the incorrect classes with $\frac{1}{N}$ instead of zero.

Label Smoothing Regularization provides small increases in performance [24], which are limited since it confuses the classifier with all classes equally. Regularizing entropy, however, confuses the classifier with its own predictions, which is informative in situations where subsets of classes are confusing, since we would want to penalize the classifier for predictions only within a subset. We argue that the mean-seeking nature of LSR omits modes in the distribution that are captured by the entropy formulation [17], i.e. we can expect an entropy regularized network to bootstrap from its own predictions in the presence of label noise, as successfully utilized by Reed *et al*. [21]. Our final objective for a batch of b training samples can be given by:

$$\mathcal{L} = \sum_{i=1}^{b} \Big(\mathcal{L}_{ce}(p_\theta(\boldsymbol{y}|\boldsymbol{x}^{(i)}), \boldsymbol{y}^{(i)}) - \frac{\alpha}{b} \cdot H(p_\theta(\boldsymbol{y}|\boldsymbol{x}^{(i)}))\Big) \tag{6}$$

3 Experiments

We continue with the formulation described in Equation 6. We experiment with implementations in popular libraries of Caffe [9] and PyTorch [18], over a cluster of NVIDIA TITAN X and GTX 1080 GPUs. We design experiments specifically to support our claim on deep learning with limited training data. We evaluate on a variety of deep learning model architectures, including AlexNet [13], VGGNet-16 [23], GoogLeNet [24] and ResNets [7]. We select the hyperparameter α via cross-validation, and present a short analysis of the effect of variation and selection of the hyperparameter on prediction performance as well.

3.1 Limited Data CIFAR-10 and CIFAR-100

Our first set of experiments include the classic image classification dataset of CIFAR-10 [12] and CIFAR-100 [12]. CIFAR-10 has 10 target classes, with 5000 samples per class, and CIFAR-100 consists of 100 classes and 500 samples per class. It is critical to note that the number of training samples per class for CIFAR-10 is much higher than that even of ImageNet [5], and hence we experiment with very small fractions of the dataset, in order to match dataset sizes present in domains with limited training data. We progressively experiment training models with randomly selected subsets of the training data and observing the gain in test performance.

As hypothesized, we observe an increase in validation accuracy with limited amounts of training data in both cases, that reduces as the amount of training data available increases. In case of CIFAR-10, we observe no significant performance increase (consistent with [19]) when trained using the entire training dataset, however, as depicted in Table 1, for small amounts of training data (1% to 10%) we observe a performance increase as large as 6%. These improvements are present across several model architectures (see Figure 1a and Table 1). We also observe that the gains reduce when using complete training data—with no significant improvement in CIFAR-10 and a gain of 1% to 3% on

Method	Accuracy(%) on CIFAR-10					Accuracy(%) on CIFAR-100				
	1%	2%	5%	10%	100%	1%	2%	5%	10%	100%
ResNet20 [7]	29.98	35.19	43.54	51.25	92.34	19.45	25.78	33.91	36.16	75.81
ResNet20 (**E**)	32.75	39.95	45.90	54.89	92.57	21.06	27.35	36.08	42.12	77.40
VGGNet16 [23]	30.18	36.02	41.78	50.21	92.06	18.19	22.35	30.16	35.46	73.81
VGGNet16 (**E**)	33.51	40.68	46.11	55.62	92.17	20.05	23.39	34.08	40.20	75.08
GoogLeNet [24]	26.15	33.57	38.10	46.18	84.16	17.90	20.38	25.34	30.19	70.24
GoogLeNet (**E**)	30.14	36.15	42.25	50.19	84.19	19.17	23.65	28.86	34.01	73.15

Table 1: The impact of adding entropy regularization on classification for random subsets of CIFAR-10 and CIFAR-100 datasets. The models trained with entropy regularization are displayed with the tag (**E**).

CIFAR-100. This can be attributed to the fact that as more training data becomes available, the training data approximates the validation data more precisely, alleviating the need to account for confusion (or penalize peakiness).

3.2 Fine-Grained Visual Classification

Fine-Grained Visual Classification (FGVC) is an important problem in computer vision, which involves distinguishing between object classes with substantially higher visual similarity compared to those in large-scale image classification tasks. Some examples of FGVC include differentiating between species of birds, flowers and animals; or the brands and models of vehicles. These tasks depart from conventional image classification in that they require expert knowledge, rather than crowdsourcing, for gathering annotations. Additionally for fine-grained wildlife data collection, several species are generally harder to photograph, resulting in long tails in the data distribution. Owing to the difficulty in capturing and annotating samples, most FGVC datasets are orders of magnitude smaller than traditional image classification datasets, making it harder to learn deep learning classifiers on such data.

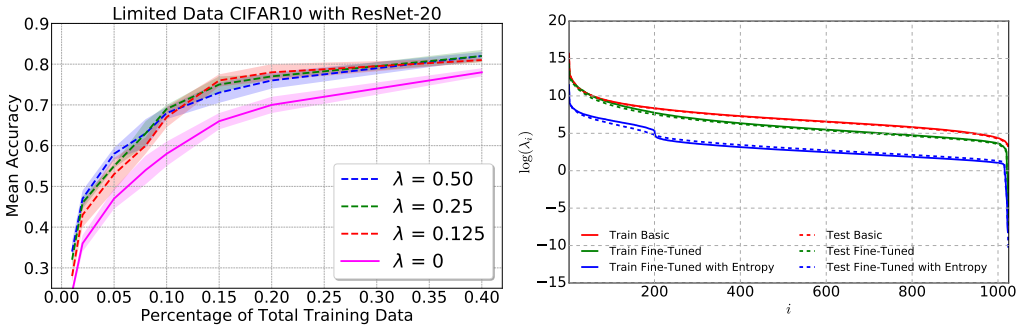
Method	Accuracy(%) on FGVC Dataset		
	CUB-2011 [26]	Stanford Dogs [11]	NABirds [1]
GoogLeNet [24]	68.19 ($\pm$ 0.14)	55.76 ($\pm$ 0.08)	70.66 ($\pm$ 0.28)
GoogLeNet (**E**)	74.37 ($\pm$ 0.19)	61.98 ($\pm$ 0.16)	71.15 ($\pm$ 0.19)
VGGNet16 [23]	73.30 ($\pm$ 0.22)	61.87 ($\pm$ 0.25)	68.29 ($\pm$ 0.31)
VGGNet16 (**E**)	77.91 ($\pm$ 0.17)	65.56 ($\pm$ 0.23)	72.66 ($\pm$ 0.23)
ResNet50 [7]	76.15 ($\pm$ 0.18)	69.92 ($\pm$ 0.16)	63.55 ($\pm$ 0.15)
ResNet50 (**E**)	81.24 ($\pm$ 0.28)	74.31 ($\pm$ 0.31)	70.84 ($\pm$ 0.18)
BilinearCNN [14]	84.10 ($\pm$ 0.12)	82.13 ($\pm$ 0.22)	80.90 ($\pm$ 0.16)
BilinearCNN (**E**)	84.93 ($\pm$ 0.15)	83.04 ($\pm$ 0.14)	81.14 ($\pm$ 0.12)

Table 2: The impact of adding entropy regularization on classification for fine-grained visual classification datasets. All models have been fine-tuned from their publicly available ImageNet-trained weights. The models trained with entropy regularization are displayed with the tag (**E**).

Our results for FGVC tasks are summarized in Table 2. We observe gains larger than the previous set of experiments, especially on generic models such as GoogLeNet [24], VGGNet16 [23] and ResNet50 [7]. We additionally observe an improvement on specialized FGVC deep models, such as Bilinear-CNN [14] across 3 datasets (CUB-2011 [26], Stanford Dogs [11], NABirds [1]).

4 Analysis and Conclusion

In this paper, we demonstrated the impact of regularizing entropy for tasks involving learning with limited amounts of training data. Since we are adding a regularizer, it would be critical to understand the variation of performance with variation in the hyperparameter. We find that for smaller amounts of data present, larger values of α provide benefits in classification, but overall we find that the algorithm is largely insensitive to different values of α in the range of 0 to 1. This variation in performance is summarized for CIFAR-10 in Figure 1a.



(a) (b)

Figure 1: (a) Analysis of variation of classification performance as training data is increased, plotted for various values of α, on CIFAR10 with model ResNet20. (b) Eigenvalue decomposition of covariance (unnormalized PCA) of penultimate layer GoogLeNet features for both training and test sets of CUB2011. We plot the value of $\log(\alpha_i)$ for the ith eigenvalue α_i obtained after decomposition of test set (dashed) and training set (solid) on three models.

Subsequently, it is also interesting to visualize what the effect of regularizing entropy is on the underlying feature maps. Adding entropy to the classifier will encourage the classifier to reduce the specificity of the features, since we discourage peakiness in the output distribution. To evaluate this hypothesis, we perform the eigendecomposition of the covariance matrix (unnormalized PCA) on the penultimate layer features of GoogLeNet trained on CUB-2011, and analyze the trend of sorted eigenvalues (Figure 2b). We examine the features obtained from a network with (i) no fine-tuning ("Basic"), (ii) fine-tuning without regularization, and (iii) fine-tuning with entropy regularization.

For a feature matrix with large covariance between the features of different classes, we would expect the first few eigenvalues to be large, and the rest to diminish quickly, since fewer orthogonal components can summarize the data. Conversely, in a completely uncorrelated feature matrix, we would see a larger tail in the decreasing magnitudes of eigenvalues. Figure 1b shows that for the Basic features (with no fine-tuning), there is a fat tail in both training and test sets due to the presence of a large number of uncorrelated features. After fine-tuning on the training data, we observe a reduction in the tail of the curve, implying that some generality in features has been introduced in the model through the fine-tuning. The test curve follows a similar decrease, justifying the increase in test accuracy. Finally, for entropy regularization, we observe a substantial decrease in the width of the tail of eigenvalue magnitudes, suggesting a larger increase in generality of features in both training and test sets, which confirms our hypothesis.

In conclusion, we demonstrate the effectiveness of regularizing entropy when training deep neural network models with limited amounts of training data. Our work should be useful in improving generalization performance of the large number of applied computer vision tasks that utilize deep neural networks for training with small amounts of data, including medical imaging, fine-grained recognition, and domain adaptation.

Acknowledgements: The authors would like to thank Ryan Farrell and Pei Guo for their helpful comments and discussions.

References

[1] Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014.

[2] Hao Chen, Xiaojuan Qi, Lequan Yu, Qi Dou, Jing Qin, and Pheng-Ann Heng. Dcan: Deep contour-aware networks for object instance segmentation from histology images. *Medical image analysis*, 36:135–146, 2017.

[3] Sumit Chopra, Suhrid Balakrishnan, and Raghuraman Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains.

[4] Brian Chu, Vashisht Madhavan, Oscar Beijbom, Judy Hoffman, and Trevor Darrell. Best practices for fine-tuning visual classifiers to new domains. In *Computer Vision–ECCV 2016 Workshops*, pages 435–442. Springer, 2016.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[9] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[10] James M Joyce. Kullback-leibler divergence. In *International Encyclopedia of Statistical Science*, pages 720–722. Springer, 2011.

[11] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs.

[12] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset, 2014.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[15] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *arXiv preprint arXiv:1702.05747*, 2017.

[16] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[17] Tom Minka et al. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

[18] Adam Paskze and Soumith Chintala. Tensors and Dynamic neural networks in Python with strong GPU acceleration. `https://github.com/pytorch`. Accessed: [August 1, 2017].

[19] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[20] Lutz Prechelt. Early stopping-but when? *Neural Networks: Tricks of the trade*, pages 553–553, 1998.

[21] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[22] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[25] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

[26] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Antonio Torralba, and Aude Oliva. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*, 2016.