Exploiting Benford's Law for Weight Regularization of Deep Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Stochastic learning of Deep Neural Network (DNN) parameters is highly sensitive to training strategy, hyperparameters, and available training data. Many state-of-the-art solutions use weight regularization to adjust parameter distributions, prevent overfitting, and support generalization of DNNs. None of the existing regularization techniques have ever exploited a typical distribution of numerical data sets with respect to the first non-zero (or significant) digit, called Benford's Law (BL). In this paper, we show that the deviation of the significant digit distribution of the DNN weights from BL is closely related to the generalization of the DNN. In particular, when the DNN is presented with limited training data. To take advantage of this finding, we use BL to target the weight regularization of DNNs. Extensive experiments are performed on image, table, and speech data, considering CNN and Transformer-based architectures with varying numbers of parameters. We show that the performance of DNNs is improved by minimizing the distance between the significant digit distributions of the DNN weights and the BL distribution along with L2 regularization. The improvements depend on the network architecture and how it deals with limited data. However, the proposed penalty term improves consistently and some CNN-based architectures gain up to 15% test accuracy over the default training scheme with L2 regularization on subsets of CIFAR 100.

1 Introduction

The advent of *Deep Neural Networks* (DNNs) has revolutionized several domains by exploiting their adaptability and robust learning capabilities without requiring full model interpretability. A critical aspect of DNNs is parameter learning, a process that is influenced by factors such as the amount of training data and hyperparameter settings. Although the number of large datasets in DNN research is constantly increasing, many industrial applications must deal with smaller, manually collected datasets. In addition, the choice of hyperparameters significantly affects the training stability and overall performance of DNNs.

A notable area of research in DNNs is weight regularization. Over the years, numerous studies have shown how explicit weight regularization described in Van Laarhoven (2017), such as L_2 or L_1 regularization, can improve learning by imposing penalties to encourage smaller or sparser weight distributions. Implicit regularization techniques, such as *Mixup* by Zhang et al. (2018) or *Cutmix* by Yun et al. (2019), modify the input data to achieve similar effects. These strategies highlight the importance of weight regularization in improving the learning capabilities of DNNs. However, they are scale dependent and need to be fine-tuned for individual domains.

This paper explores a remarkable aspect of DNNs - the distribution of the first significant digits of their weights, a phenomenon closely related to Benford's Law (BL). Discovered independently by Newcomb (1881) and Benford (1938), BL describes a counterintuitive but common pattern in numerical data across several domains: Lower significant digits occur more often than higher ones.

This scale-invariant pattern described by BL has been observed in various domains, such as physical constants by Shao & Ma (2010) and stock prices by Ley (1996). In machine learning, the properties of BL have been exploited to detect anomalies in input data by O'Mahony et al. (2023) and synthetically generated images by Bonettini et al. (2021), underlining its relevance in DNN research. Their motivation stems from

the observation that natural datasets tend to follow BL and synthetic data do not. Sahu et al. (2021) show that the link between BL and DNNs is based on their mutual relationship to thermodynamics. In particular, the probability of energy states in closed systems, where smaller energy states are more likely to occur than larger ones. The same pattern as for BL.
Despite the highlighted correlations between DNNs and the Benford distribution, ML research has only focused on observing whether the collected significant digits of trained weights follow BL or not. In particular, the authors of Sahu et al. (2021) propose to use the correlation between significant digits of DNN weights and BL to estimate the generalization error of DNNs. A high correlation is then used as a stopping criterion for *Early Stopping*, a mechanism that stops training when performance on a holdout validation dataset does not increase for a predefined number of consecutive epochs. This in turn eliminates the need for a validation dataset, which can be used for training instead.
This feature of BL is closely related to its relevance for bias detection in numerical data such as synthetic images. In particular, data, especially with limited quantity, is always biased because it represents a small snapshot of the real world. In the same way, this bias is reflected in the weights of DNNs. Thus, we expect a DNN trained with limited training data to deviate from BL. In addition, biasing the weights towards BL with a penalty added to the loss function should improve its performance.
To the best of the authors notice, there is currently no research available that promotes BL as a regularization technique. To analyze the effects of BL on DNNs weights regularization, this paper:

1. demonstrates the relation between ML datasets and BL
2. introduces a way to approximate BL via gradient-based optimization,
3. analyzes the effect of Benford regularization in training DNNs with reduced datasets compared to their standard L2 regularized training scheme.

Benford regularization is of particular interest for applications where either a complex dataset is too small for a DNN or a DNN has limited capacity for a large dataset. To this end, our experiments use subsets of common image (CIFAR10/100) and audio datasets to emphasize the bias in the dataset. In that way, we compare CNN-based architectures such as DenseNet and ResNext and Transformer architectures for small datasets like tiny ViT and tiny Swin Transformer with respect to performance on random subsets.
Since L2 regularization is the most commonly used penalty term in DNN training and is also part of the originally proposed training scheme, we use L2 regularization as the baseline. The results show consistent improvements of the Benford regularization in combination with L2 regularization over L2 regularization alone. In particular, CNN-based architectures benefit from additional Benford regularization in subsets of the datasets with improvements of up to 15%. The Transformer models also improve with Benford and L2 regularization on the full dataset up to 3%. Additionally, we experiment with small and hardware-optimized MobileNetv3 models on the Imagenet1K dataset, where the larger capacity model improves more with Benford regularization as the smaller one. The experiments show an interplay of improved performance with Benford regularization , model capacity and data availability.
The remainder of the manuscript is structured as follows. First, we investigate related research on weight regularization and BL in connection to DNNs. Afterward, we outline the relation between BL and physics and illustrate a close connection to DNNs. The proposed method introduces a differentiable approximation of the Benford distribution and how BL can be approximated via gradient descent. The following experiments are performed on the public image datasets CIFAR10/100 by Krizhevsky (2009) and Imagenet1K by Russakovsky et al. (2015) and evaluate the performance of well-known DNNs trained from scratch with different dataset sizes. To complete the experiments, the regularizer is evaluated on different data domains, such as speech and tabular data. After discussion of the experiments and their results, we assess the limitations of the presented work. Finally, we present the conclusion and outlook of the paper.

2 Considered problem and related work

Research on weight regularization focuses on specific favorable weight distributions. The goal of these approaches is to make DNN training stable while ensuring robustness against overfitting and noise in the data. This section discusses state-of-the-art research that employs regularization and data augmentation for DNNs, and related work that features BL with respect to DNNs.

Regularization and data augmentation are of practical importance when training DNNs to avoid overfitting and foster numerical stability. Weight regularization, like L_2 or L_1, aims to reduce the norm of the weights by adding a penalty to the loss function. L2 regularization is mainly implemented as weight decay as described in Loshchilov & Hutter (2019). Alongside L2 regularization, the focus of seminal work is on implicit regularization, such as *data augmentation* methods reviewed by Shorten & Khoshgoftaar (2019) like *Mixup* by Zhang et al. (2018) or *Cutmix* by Yun et al. (2019). Such techniques do not directly constrain the weights, but they affect their distribution. *Data augmentation* distorts the input data to make the network robust against different conditions, such as lighting conditions for camera images, thus improving generalization. To achieve this, *Mixup* fuses two training samples and their respective label via a convex combination to learn smooth decision functions. *Cutmix* extends the approach and cuts out a patch from an input image and replaces the patch with information from a different training sample. The targets are adjusted proportionally to number of switched pixels. An alternative regularization method, *SAM* by Foret et al. (2019), reduces sharpness of the loss landscape via second order gradient smoothing. Generalization is achieved by finding parameters that lie in a neighborhood with a low loss instead of suboptimal sharp minima. At last, *Early Stopping* is one of the most common methods in DNN research. It describes a mechanism to stop the training when a monitored metric, like validation error, stops improving Yao et al. (2007). In particular, *Early Stopping* does not change the weights by adding a penalty or changing the input data, so it is of relevance for training DNNs in any data domain and is commonly used. In the work of Sahu et al. (2021), the deviation to BL has been used as metric when to stop the training. This metric makes a validation dataset obsolete, which in turn can be used as additional training data.

The distribution of significant digits has not yet been used to regularize the weights of DNNs via gradient-based optimization. So far, BL has been used to analyze the characteristics and the liability of the data. In Bonettini et al. (2021), the authors use the significant digits of the cosine transformation coefficients of an image to detect whether it was generated by a *Generative Adversarial Network* (GAN). Similarly, BL has been used in O'Mahony et al. (2023) to discriminate between natural and corrupt data, acting as a filter to detect out-of-distribution data or anomalous data points. These methods monitor whether the collected data or generated images follow BL and are motivated by the observation that natural datasets are known to obey BL. Recently, Sahu et al. (2021) showed that the distribution of significant digits is a predictor of the generalization of DNNs to the validation data and can be used as *Early Stopping* criterion. Until now, BL has not been incorporated into the optimization process, thus missing important features of the Benford distribution.

3 Motivation

Regularization and data augmentation are two practical techniques that can significantly improve the training and generalization of DNNs. However, these techniques need to be tailored to the specific data being used, which can result in a non-universal approach. In various domains, researchers have observed that the significant digits of datasets often follow a peculiar pattern, as shown in Figure 1, known as BL. Despite its widespread occurrence, the underlying reason behind BL remains a mystery Wang & Ma (2023). Interestingly, random samples from randomly selected distributions have been shown to converge to BL Hill (1995). This finding is also supported by the original work of Benford, who noted that while some datasets deviate from BL, their union closely follows it Benford (1938), underlining that unbiased data tend to obey BL. To further illustrate this phenomenon, we analyzed the frequency spectrum of several image datasets commonly used for DNN training. The coefficients of the Discrete Cosine Transformation (DCT) and Fast Fourier Transformation (FFT) are known to follow BL, see Benford (2021). As shown in Figure 1, not all frequencies of these datasets follow BL. However, the union of all datasets gets very close, providing further evidence that data from various unbiased distributions obey BL. Mathematically, the law is fulfilled when the frequency $P(d)$ of any significant digit d is given as

$$P(d) = log_b \left(1 + \frac{1}{d}\right), \tag{1}$$

where b is the base of the number system. Alternatively, the authors in Berger & Hill (2015) demonstrate that a sequence satisfies BL if and only if "the fractional parts of its decimal logarithm are uniformly distributed between zero and one", as shown in Eq.2

$$P(d) = log_{10} \left(1 + \frac{1}{d}\right) \quad iff \quad log_{10}(X) \; mod \; 1 \; \sim \mathcal{U}(0,1), \tag{2}$$

where X represents arbitrary numerical data. The described law is of particular interest because it is scale- and base-invariant, as shown by Berger & Hill (2021). To measure the deviation from BL, we compute the Kullback-Leibler (KL) divergence by Kullback & Leibler (1951) between the significant distribution of the weights and BL (BL KL), formally defined in Eq. 3

$$KL(Q \parallel BL) = \sum_{d=1}^{9} Q(d) \log \left(\frac{Q(d)}{P(d)}\right), \tag{3}$$

where $Q(d)$ is the observed frequency of the d-th significant digit in the DNN weights. In the experimental section, we show that networks trained with more data have weights closer to BL.

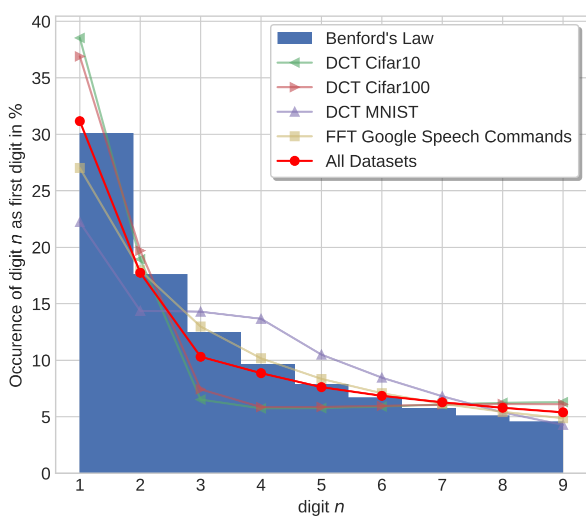


Figure 1: Comparison of significant digit distribution of DCT/FFT coefficients in public image and speech datasets, the union of datasets, and Benford's Law.

These observations are related to theoretical results from Iafrate et al. (2015) on the partitioning of numbers, a topic of number theory, and its relation to BL.

A partitioning process describes how a fixed quantity is distributed. For example, partitioning describes the energy states of particles in closed systems, see Iafrate et al. (2015) and is used in combinatorics to describe the partitioning of an integer into positive smaller integers. Common DNN structures like CNNs use fixed filters that partition the input and Transformers extract information from partitioned input data. Formally, we define a quantity X that is divided into smaller parts:

$$X = \sum_j n_j x_j, \tag{4}$$

with parts n_j of size x_j. The goal is to find the distribution $p(n)$ that explains how often we observe a part of size x in the subset that adds up to X. Intuitively, it is obvious that smaller numbers occur more often, which can be easily verified by inspecting the subsets that add up to ten: $[\{9,1\},\{8,2\},\{8,1,1\},...,\{1,...,1\}]$. An illustrative example from Kafri (2009) describes the system of Eq. 4 with a number X, represented as "boxes" with N non-interacting balls. The question is how many boxes have exactly n balls, which in turn gives information about the probability of integers in a number. The combination of different boxes then forms a number. As the probability of a ball being in a box is the same for all boxes, the distribution of balls obeys BL. This result is further generalized by Iafrate et al. (2015), showing that the distribution of parts approximately follows a simple power law $p(n) \sim 1/x$. The significant digit distribution P_d of the inverse power distribution obeys BL, as shown in Eq. 5

$$P_d = \frac{\int_{d10^p}^{(d+1)10^p} dx/x}{\int_{10^p}^{10^{p+1}} dx/x} = \frac{\ln(1+1/d)}{\ln(10)} = \log_{10}\left(1+\frac{1}{d}\right). \tag{5}$$

Both approaches of Iafrate et al. (2015) and Kafri (2009) assume equal probabilities for each ball to fall in a box or the parts of each size. This assumption is based on the maximum entropy principle by Jaynes (1957). Whenever no further information about a probability distribution is given, the maximum entropy principle is the most unbiased assumption. Finding probability distributions that maximize the entropy is used in Machine Learning research for regularization in Haarnoja et al. (2018) and Chiang et al. (2005). Like partitioning, DNNs for classification follow a top-down strategy that divides the input into a smaller feature space by partitioning it with the trained weights. In DNNs, the number of features represents the parts, and the weights represent their respective sizes. With sufficient training data, we expect an unbiased partitioning of the input space into the feature space, resulting in weights that are approximately Benford distributed. However, in the absence of training data, the parts are biased towards fewer features with larger weights, leading to a deviation from BL. This provides an explanation for the data obtained from Figure 2, where the lack of information in the training data leads to less diversity in the feature space and, thus, to an increase in the distance to BL.
This motivates to find a way to exploit BL for weight regularization in DNNs. Specifically, our goal is to constrain the significant digit distribution of the weights closer to BL in order to improve the test error independently of the task, focusing on smaller subsets of datasets. To the best of our knowledge, this is the first method that incorporates BL into the optimization process of neural networks via gradient-based optimization.

4 Approach

This section presents a framework for incorporating BL into neural network regularization. To this end, we propose a differential approximation of BL that can be used to optimize the significant digit distribution of the DNN weights. The significant digits of the weights are updated via gradient descent to close the distance to BL. To further improve the proposed approach, we use derived error functions on BL for exponential functions that relax the optimization problem.
The approximation of BL is more tractable with the right-hand side of Eq. 2 as it depends on the input data x and not on the significant digits. The modulo 1 operator is the same as computing the fractional part of a number. The fractional part of any positive number x is defined as

$$frac(x) = x - \lfloor x \rfloor, \tag{6}$$

where $\lfloor \cdot \rfloor$ is the floor function and $\lfloor x \rfloor$ denotes the next lower integer value of x. This function has discontinuities at zero and one, but the gradient can be numerically approximated in those regions. To approximate the uniform distribution of the fractional part with gradient optimization, we utilize quantile regression described in Koenker & Hallock (2001). Quantile regression has been used lately in Reinforcement Learning (RL) to learn arbitrary distributions, like Dabney et al. (2018). Specifically, we compare the quantiles of the neural network weights to the quantiles of a uniform distribution $\mathcal{U}(0,1)$. In terms of the cumulative distribution function (c.d.f.), the quantile p defines the probability that a random variable X evaluates smaller or equal to a threshold x, as shown in Eq. 7

$$F_X(x) = Pr(X \leq x) = p. \tag{7}$$

The quantile function $Q(p)$ is defined as the inverse c.d.f, presented in Eq. 8.

$$Q(p) = F_X^{-1}(p) = \begin{cases} x_{(k)}, & k = n^{(p)} \text{ if } n^{(p)} \text{ is an integer} \\ \frac{1}{2}(x_{(k)} + x_{(k+1)}), & k = \lfloor n^{(p)} \rfloor \text{ if } n^{(p)} \text{ is not an integer} \end{cases} \tag{8}$$

Here, $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$ denote the ordered weights in ascending order and n is their total number. The $\lfloor \cdot \rfloor$ denotes the floor function. The expression $n^{(p)}$ is the index of the p-th quantile, and the value of k is the integer part of $n^{(p)}$. If $n^{(p)}$ is not an integer, the p-th quantile is defined as the average of the k-th and $(k+1)$-th ordered observations. The p-th quantile of the standard uniform distribution $\mathcal{U}(0,1)$ is already given by p. To measure the deviation from the standard uniform distribution, we compare the quantiles of the weight distribution to the quantile value, as shown in Eq. 9

$$L_{BL}(K,\theta) = \frac{1}{K}\sum_{k=1}^{K}\left(\hat{Q}(k,\theta) - k\right)^2, \tag{9}$$

where K is the number of quantiles and $\hat{Q}(k,\theta)$ denotes the k-th quantile of the weights θ. The number of quantiles used for the regression is essential for an accurate approximation of the standard uniform distribution. The best estimation can be achieved when the number of quantiles is equal to the number of weights in the DNN. Quantiles are computed based on the Quickselect algorithm by Hoare (1961) which has an average complexity of $\mathcal{O}(n)$. This is infeasible for DNNs with millions of parameters. Thus, whenever needed, the quantile regression is computed for each layer sequentially. This loss formulation is closely related to isotonic regression for DNN calibration Niculescu-Mizil & Caruana (2005).

In Algorithm 1, we illustrate a Pytorch-style pseudocode of the quantile regression steps. Similarly to other regularizers, the quantile regression loss is added to the objective function. The approach is consequently not limited to classification tasks, but can be used in any optimization process.

However, to improve the stability of the proposed loss, we incorporate the error function from Engel & Leuenberger (2003). They show that exponential functions oscillate around BL. The unnormalized probability density of a classification network is defined as:

$$p(x,\theta) = e^{(f(x,\theta)/\tau)}, \tag{10}$$

where $f(x,\theta)$ defines the neural network output and τ is the temperature. In Engel & Leuenberger (2003) the authors show that exponential functions of the form $\lambda e^{-\lambda x}$ obey BL within error bounds independent of λ and x. The error on BL for the exponential function is defined as:

$$Er(f) = \sum_{-\infty}^{+\infty} e^{-\lambda d 10^n}(1 - e^{-\lambda 10^n}) - \log_{10}(1 + \frac{1}{d}). \tag{11}$$

According to the calculation in Engel & Leuenberger (2003), the error is bound within $Er(f) \leq 0.03$. This result can be incorporated into the proposed Benford regularizer, such that losses within the error bound are neglected.

Algorithm 1: PyTorch-style pseudocode of computing the quantile loss for Benford regularization

```
# W - flattened model weights of shape Nx1
def quantile_loss_BL(W):
   # Number of quantiles is equal to the number of weights N
   n_quantiles = W.shape[0]
   # Compute fractional part
   W = remainder(log10(W),1)
   # Define quantile steps between 0 and 1
   quantile_steps = linspace(start=0, end=1, steps=n_quantiles)
   # Compute quantiles for the weights
   W_quantiles = quantile(W, quantile_steps)
   # Uniform quantiles are the quantile steps
   uniform_quantiles = quantile_steps
   # Compare the quantiles with Mean Squared Error
   bl_loss = mse_loss(W_quantiles, uniform_quantiles)
return bl_loss
```

5 Experiments

In this section, we assess the Benford regularization on different datasets and model architectures. The baseline are networks trained with L2 regularization since this is part of their originally proposed training scheme. The Benford regularizer is evaluated on top of the L2 regularization which aids for numerical stability.
To this end we train all networks from scratch on subsets of publicly available image datasets such as MNIST (LeCun (1998)), CIFAR 10/100 (Krizhevsky (2009)) and ImageNet1K (Russakovsky et al. (2015)). In order to show invariance to different model architectures, we evaluate the Benford regularizer on CNN and Transformer models. The Transformer models used throughout the experiments have 3 times more parameters than the CNN-based networks, making them prune to overfit on small datasets. At the same time, they have a larger capacity which is relevant when the experiments use the full dataset. Besides images, the experiments evaluate the performance of the M5 model from Dai et al. (2017) on the Google Speech Commands dataset (Warden (2018)) and a two-layer MLP on the tabular Iris dataset (Fisher (1936)). The hyperparameters used for training are defined in the respective sections. All reported results for CIFAR 10/100 are obtained from 15 different seeds and the mean and standard deviation is reported in the respective tables. The results from ImageNet1K are obtained from 3 different seeds. We illustrate the mean and standard deviation on the validation performance. Afterward, the table presents the test accuracy of the model with the highest validation performance.
The implementation is based on Pytorch™ v2.0 Paszke et al. (2019) and as processing unit, we used two NVIDIA® Tesla® A30 GPUs. The implementation is publicly available. [1]

5.1 MNIST experiments

To show that less training data leads to a larger KL divergence and thus a larger deviation from BL, we evaluated LeNet Lecun et al. (1998) in combination with Benford and L2 regularization on different data subsets. The Benford regularizer is scaked by 0.1 and we presented the average test error curves across 5 seeds in Figure 2. The results highlight that fewer data leads to more improvement that can be achieved with Benford regularization. Besides the clearly improved test performance when training only on 1% of the training data, the improvement decreases with more training data. Regarding the deviation to BL, the proposed regularization successfully reduces the BL KL but the regularization cannot compensate for the missing data. The network with lower BL KL shows superior performance. Especially, when using 10% and 1% of the data, the BL regularized network converges faster. However, the BL KL becomes similar for both networks during training. This shows a clear connection between the BL KL and test error and the importance of the proposed regularizer for convergence speed.

5.2 CIFAR experiments

We first show the effectiveness of the Benford regularizer on the CIFAR 10 and CIFAR 100 datasets. Both consist of 50,000 training images and 10,000 test images with 10 and 100 different classes respectively. A subset of 10,000 images from the training data is used for validation. During the evaluation of different dataset sizes, the validation and test data size stays the same. To this end, we present the results of CNN- and Transformer-based models introduced by Vaswani et al. (2017). For this, the DenseNet by Huang et al. (2017) with depths 121 and ResNext with 29 layers by Xie et al. (2017) represent the CNN family. Furthermore, the tiny version of the Swin Transformer by Liu et al. (2021) and the Vision Transformer (ViT) for small datasets proposed by Lee et al. (2021) represent the Transformer models. The input images are normalized 32×32 images with a random crop and random horizontal flip during training. For testing, the images are only normalized. We use an SGD optimizer with 0.9 momentum and $5 \cdot 10^{-4}$ weight decay and an initial learning rate of 0.001 for the CNN models and the Adam optimizer by Kingma & Ba (2014) with an initial learning rate of 0.0001 with a cosine annealing learning rate schedule for the Transformer models. Each network is trained for 200 epochs and the learning rate of the CNNs is divided by 5 after $[60, 120]$ epochs, following the settings in DeVries & Taylor (2017). For all models, we compute the quantile regression loss for each layer and add the average as regularization with a scaling factor of 0.1. Figure 3 illustrates that the

[1] The code is published on Github upon acceptance.

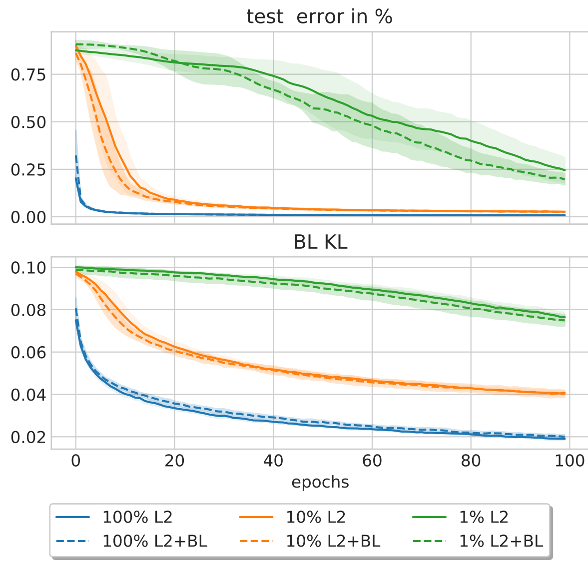


Figure 2: Comparing the test error in % and BL KL of L2 and L2 with Benford regularization when training LeNet on 100%, 10% and 1% of the MNIST dataset.

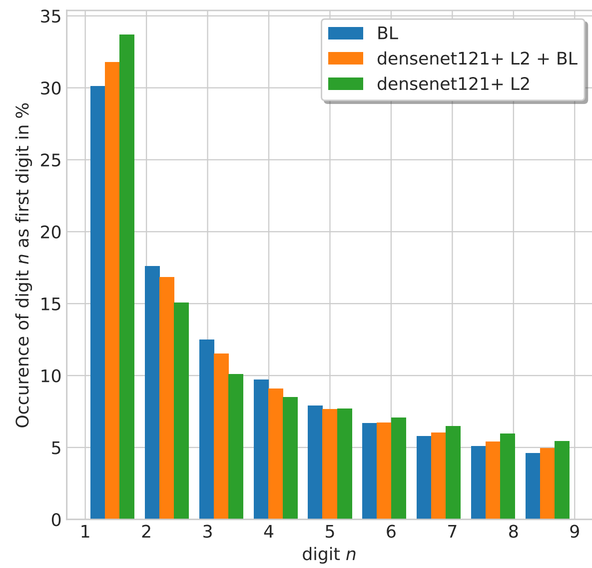


Figure 3: Comparison between Benford's Law and significant digits of DenseNet121 weights trained with L2 and Benford regularization and with L2 regularization alone on CIFAR 10.

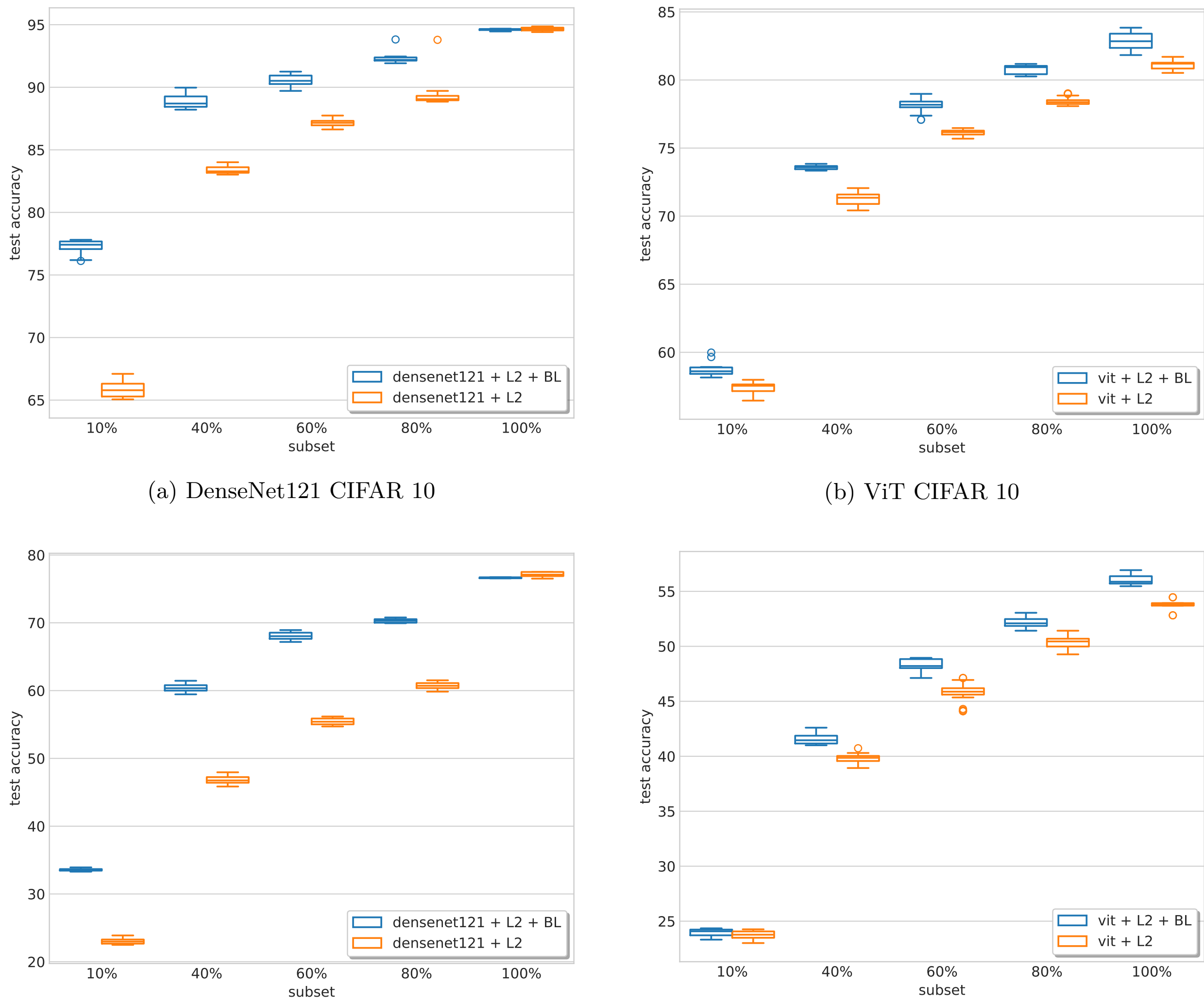
proposed Benford regularizer not only minimizes the BL KL but tweaks the significant digits of the DNN weights towards BL, while the same network optimized with L2 regularization only deviates from BL. The results in Table 1 and Table 2 report the mean and standard deviation of the test error obtained over 15 seeds. The results show that the proposed Benford regularizer improves the network performance when the number of data samples is limited. As an additional illustration, Figure 4 show the improvements of the tiny ViT's and DenseNet121's test performance on the subsets of CIFAR 10 and CIFAR 100 with the additional use of Benford regularization.

We want to highlight that the ViT trained with L2 and Benford regularization on 80% of CIFAR 10, achieves similar performance as the ViT trained on the full dataset. In general, the ViT benefits from the L2 and Benford regularization on all dataset sizes. Since, the Transformers are much larger than the tested CNN models, the overall test accuracy is lower but the disparity between model size and dataset diversity explains why the ViT benefits on subsets and the full dataset. On the other end, the effect of Benford regularization vanishes when the dataset is too complex for the available amount of data and model complexity, as shown by the performance of ViT on 10% of CIFAR 100.

In contrast, the CNN models improve up to 15% on the subsets of CIFAR 100 and up to 10% on subsets of CIFAR 10. In addition, the Benford regularization has limitations on the full dataset for the CNN models, where their performance is near optimum for the respective architecture.

5.3 Imagenet experiments

In the last set of image classification experiments, we increased the complexity of the task using the ImageNet1K (Russakovsky et al. (2015)) dataset. In this set of experiment, the focus is on the MobileNetv3 networks by Howard et al. (2019) that are optimized for mobile applications. In contrast to the previous experiments, this section focuses on reduced models for large datasets. To this end, we train the small and large version of MobileNetv3 on the Imagenet1K dataset with default data augmentation, like horizontal flipping, random cropping and color jittering. We scale the Benford regularization loss with a factor of 10^{-6}, which gives the best results. The networks were trained for 90 epochs, which was the convergence limit, with an initial learning rate of 0.1 which was divided by 10 after 30 and 60 epochs. Due to limited resources we train the networks with a batch size of 256 images, while the networks proposed in Howard et al. (2019)



(a) DenseNet121 CIFAR 10

(b) ViT CIFAR 10

(c) DenseNet121 CIFAR 100

(d) ViT CIFAR 100

Figure 4: The distribution of DenseNet121's test accuracy in % (↑) trained on subsets of CIFAR 10 (a) and CIFAR 100 (c) shows that it improves with BL regularization on all subsets but achieves the same performance as L2 regularization on the full dataset, respectively. The distribution of tiny ViT's test accuracy on CIFAR 10 (b) and CIFAR 100 (d) improves on the subsets and on the full dataset over L2 regularization.

Table 1: Average test error in % (↓) and standard deviation on CIFAR 10. An evaluation of the L2 regularization with (L2+BL) and without Benford regularization on the full dataset and random subsets of 80%, 60%, 40%, and 10%. We highlight the respective results in bold when the performance difference is at least one standard deviation.

		Subset				
	Reg.	100%	80%	60%	40%	10%
Swin	L2	15.70 ± 0.35	17.44 ± 0.48	20.39 ± 0.46	24.61 ± 0.55	42.72 ± 0.72
Transformer	L2+BL	15.17 ± 0.42	16.66 ± 0.59	19.94 ± 0.41	24.55 ± 0.50	41.72 ± 0.53
Tiny ViT	L2	18.91 ± 0.35	21.21 ± 0.35	23.86 ± 0.22	28.48 ± 0.41	42.34 ± 0.59
	L2+BL	$\mathbf{17.97 \pm 0.40}$	$\mathbf{19.51 \pm 0.31}$	$\mathbf{21.66 \pm 0.22}$	$\mathbf{26.6 \pm 0.36}$	$\mathbf{41.75 \pm 0.59}$
DenseNet121	L2	5.61 ± 0.24	10.74 ± 0.81	12.86 ± 0.27	16.30 ± 0.80	33.65 ± 0.47
	L2+BL	5.71 ± 0.245	$\mathbf{7.6 \pm 0.61}$	$\mathbf{9.01 \pm 0.27}$	$\mathbf{12.04 \pm 0.74}$	$\mathbf{27.62 \pm 0.64}$
ResNext29	L2	6.03 ± 0.19	12.88 ± 2.85	16.55 ± 0.35	22.10 ± 0.55	43.23 ± 0.39
2x64	L2+BL	6.03 ± 0.11	$\mathbf{9.04 \pm 0.29}$	$\mathbf{10.49 \pm 0.38}$	$\mathbf{13.3 \pm 0.55}$	$\mathbf{32.35 \pm 0.39}$

Table 2: Average test error in % (↓) and standard deviation on CIFAR 100. An evaluation of the L2 regularization with (L2+BL) and without Benford regularization on the full dataset and random subsets using 80%, 60%, 40% and 10% of the training data. We highlight the respective results in bold when the performance difference is at least one standard deviation.

		Subset				
	Reg.	100%	80%	60%	40%	10%
Swin	L2	45.96 ± 0.44	50.33 ± 1.82	54.94 ± 1.47	61.11 ± 1.13	78.08 ± 0.40
Transformer	L2+BL	$\mathbf{44.08 \pm 0.47}$	$\mathbf{48.09 \pm 0.46}$	$\mathbf{52.41 \pm 0.77}$	$\mathbf{58.87 \pm 0.70}$	77.65 ± 0.34
Tiny ViT	L2	46.26 ± 0.53	49.64 ± 0.52	54.21 ± 0.81	60.2 ± 0.40	76.33 ± 0.43
	L2+BL	$\mathbf{43.88 \pm 0.36}$	$\mathbf{48.43 \pm 0.58}$	$\mathbf{51.64 \pm 0.80}$	$\mathbf{57.89 \pm 0.27}$	76.28 ± 0.47
DenseNet121	L2	22.71 ± 0.70	39.25 ± 0.47	44.55 ± 0.48	53.18 ± 0.49	77.09 ± 0.60
	L2+BL	23.36 ± 0.08	$\mathbf{29.7 \pm 0.23}$	$\mathbf{32.88 \pm 0.46}$	$\mathbf{39.51 \pm 0.64}$	$\mathbf{66.43 \pm 0.72}$
ResNext29	L2	22.97 ± 0.14	43.02 ± 0.33	49.81 ± 0.39	59.11 ± 0.44	80.38 ± 0.31
2x64	L2+BL	$\mathbf{22.62 \pm 0.13}$	$\mathbf{29.99 \pm 0.35}$	$\mathbf{35.28 \pm 0.9}$	$\mathbf{42.61 \pm 0.46}$	$\mathbf{74.5 \pm 0.37}$

are trained with a batch size of 4096. In Figure 7 we illustrate the mean and standard deviation of the validation error over three runs. The best validation performance of each network was used for testing. The test accuracy in Table 3 and the validation error curve show a significant improvement of the proposed regularization method.

Table 3: Imagenet1K test accuracy in % (↑) for the MobileNetv3 networks.

	Regularization	
MobileNetv3 Large	L2	66.96
	L2+BL	**67.82**
MobileNetv3 Small	L2	59.3
	L2+BL	**59.77**

5.4 Speech and tabular datasets

To further demonstrate the effectiveness of the Benford regularizer, we propose the evaluation of different data domains. For this purpose, we train the M5 model of Dai et al. (2017) on the Google Speech Com-

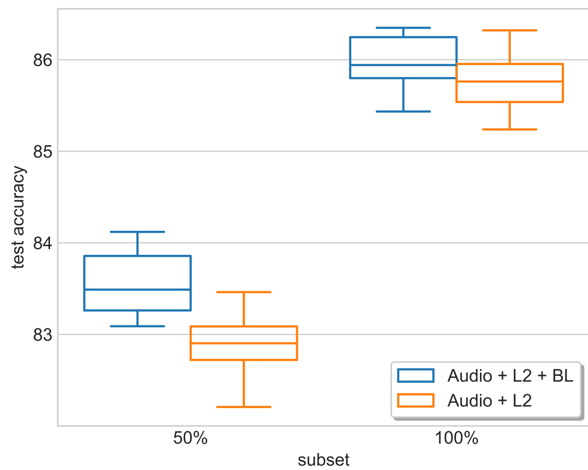


Figure 5: Distribution of the test accuracy in % (↑) trained on 100% and 50% subsets of Google Speech Commands (Audio) dataset.

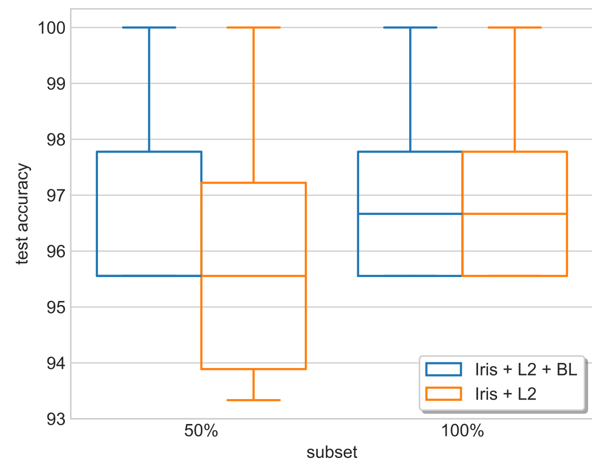


Figure 6: Distribution of the test accuracy in % (↑) trained on 100% and 50% subsets of the tabular IRIS dataset.

mands dataset (Warden (2018)). The datasets contain recordings for a total of 85,511 training, 10,102 validation, and 4,890 test recordings. For training, we use the Adam optimizer with a weight decay of 0.01 and an initial learning rate of 0.1 and scale the Benford regularizer by a factor of 0.1. The model is trained for 100 epochs, and the learning rate is divided by 10 after 20 epochs, as described in Dai et al. (2017). For the experiments on the IRIS dataset Fisher (1936), we use a neural network consisting of two fully connected layers with 10 and 3 neurons and a ReLu activation function (MLP). The dataset consists of 150 samples, randomly divided into 73 for training, 32 for validation, and 45 for testing. The model is trained for 1000 epochs with the Adam optimizer and an initial learning rate of 0.001. The Benford regularization is scaled by a factor of 0.001. The results in Table 4 show that the Benford regularization also improves the performance of DNNs regardless of the data domain. Looking at Figure 5, in the case of the more complex audio dataset, the Benford regularization improves the average performance of the model regardless of the dataset size. The gain when trained on only 50% of the audio data is significant with 1% on average and a smaller variance. For the simpler IRIS dataset, Figure 6 shows that the 50% subset performance improves with Benford and L2 regularization, while the performance is almost identical for the full dataset. Here, the network is near optimum.

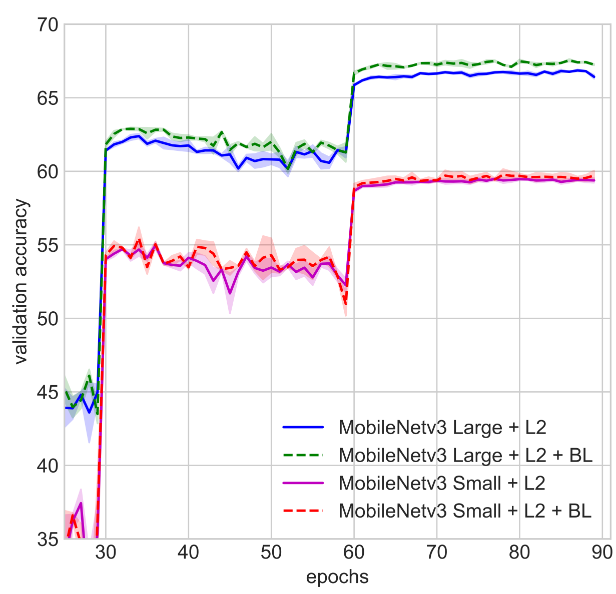


Figure 7: Validation accuracy (↑) of MobileNetv3 small and large. Both models benefit from the Benford regularization.

6 Limitations

The experimental results clearly highlight how regularizing the DNN weights towards BL improves the overall performance. Especially when the training data is limited, the proposed regularization consistently improves in addition to L2 regularization. In Figure 2 and Figure 3 we have seen that the network

weights can be optimized such that they are almost identical to BL. Since BL is scale invariant, there are infinitely many ways to approximate BL, which has been the reason to use L2 and Benford regularization together. Further, the experiments have shown that the Benford regularization is bounded by the amount of data, data complexity and model capacity. In general, when the model fails to learn reasonable features, the Benford regularization is not able to compensate that. Additionally, when the network is presented with sufficient data and it is near optimum, the Benford regularization has limited effect on distribution of the weights, because it is already closely following BL.
Since the applied error bounds are related to exponential functions, such as the softmax function, the experiments were carried out on classification tasks. Energy-based models learn an exponential energy function, which is used for generative modeling. This indicates that our approach is transferable to generative modeling but not to regression problems. Here, further analysis remains to be done.

Table 4: Average test error (↓) and standard deviation on tabular and speech datasets. An evaluation of the Benford regularization on the full datasets and a random subset of 50%.

		Subset	
	Regularization	100%	50%
M5 (Audio)	L2	14.12 ± 0.39	17.14 ± 0.38
	Benford	14.11 ± 0.31	$\mathbf{16.04 \pm 0.35}$
MLP (IRIS)	L2	3.43 ± 0.049	5.5 ± 0.05
	Benford	$\mathbf{3.12 \pm 0.046}$	$\mathbf{4.22 \pm 0.03}$

7 Conclusion and future work

While previous approaches only observed whether obtained data follows BL, this paper is the first to propose the Benford regularization, where BL is learned via gradient-based optimization. The motivation is based on commonalities between DNNs, BL and thermodynamics and the intriguing features of BL. It is scale invariant, thus independent of the data domain and is commonly used to detect bias and anomalies in measured datasets. The proposed method applies Benford regularization in combination with L2 regularization for numerical stability and presents substantial improvements. Extensive experiments were conducted on random subsets of common image datasets CIFAR 10/100 with CNN and Transformer-based models of varying number of parameters. The addition of Benford regularization boosted the performance up to 15% on subsets of CIFAR 10 and up to 10% improvement on CIFAR 100 subsets. The limitations are observed when the model either reaches the optimum or the dataset is too complex for the limited amount of data. In these cases, the performance is the same as training with L2 regularization alone. Consequently, experiments on low-capacity MobileNetv3 models are evaluated on the large and complex ImageNet1k dataset. As expected, the larger capacity network benefits more from the additional Benford regularization as the smaller version due to the larger amount of available data. This observation is further shown by additional experiments on the tabular Iris and Google Speech commands datasets.
To summarize, our experiments demonstrate an interplay between model and dataset complexity, revealing that Benford regularization yields the most significant improvements when a large-capacity model is paired with limited data or a lower-capacity model is paired with abundant data. However, we observe that there are limits to this approach, as overly complex models for simple data (e.g., Swin Transformer) or low-capacity models for complex data (e.g., MobileNetv3 Small) cannot effectively leverage this regularization.
These results not only provide insights about model capacity but have practical implications for industrial DNN applications where data collection is challenging and model sizes are constrained by hardware design.

References

Frank Benford. The law of anomalous numbers. *Proceedings of the American philosophical society*, pp. 551–572, 1938.

Frank Benford. Base dependence of benford random variables. *Stats*, 4(3):578–594, 2021.

Arno Berger and Theodore P. Hill. *An Introduction to Benford's Law*. Princeton University Press, 2015. URL `http://www.jstor.org/stable/j.ctt1dr35m0`.

Arno Berger and Theodore P Hill. The mathematics of benford's law: a primer. *Statistical Methods & Applications*, 30:779–795, 2021.

Nicolo Bonettini, Paolo Bestagini, Simone Milani, and Stefano Tubaro. On the use of benford's law to detect gan-generated images. In *2020 25th international conference on pattern recognition (ICPR)*, pp. 5495–5502. IEEE, 2021.

Yun-Wei Chiang, Peter P. Borbat, and Jack H. Freed. Maximum entropy: A complement to tikhonov regularization for determination of pair distance distributions by pulsed esr. *Journal of Magnetic Resonance*, 177(2):184–196, 2005.

Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 421–425. IEEE, 2017.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Hans-Andreas Engel and Christoph Leuenberger. Benford's law for exponential random variables. *Statistics and Probability Letters*, 63(4):361–365, 2003. ISSN 0167-7152. doi: https://doi.org/10.1016/S0167-7152(03)00101-9.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2): 179–188, 1936.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *9th International Conference on Learning Representations, ICLR 2021, Vienna, Austria, May 3-7, 2021*, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Theodore P. Hill. A Statistical Derivation of the Significant-Digit Law. *Statistical Science*, 10(4):354 – 363, 1995. doi: 10.1214/ss/1177009869.

C. A. R. Hoare. Algorithm 65: find. *Commun. ACM*, 4(7):321–322, 1961. ISSN 0001-0782. doi: 10.1145/366622.366647. URL `https://doi.org/10.1145/366622.366647`.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Joseph R Iafrate, Steven J Miller, and Frederick W Strauch. Equipartitions and a distribution for numbers: A statistical model for benford's law. *Physical Review E*, 91(6):062138, 2015.

E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957. doi: 10.1103/PhysRev.106.620. URL `https://link.aps.org/doi/10.1103/PhysRev.106.620`.

Oded Kafri. Entropy principle in direct derivation of benford's law. *arXiv preprint arXiv:0901.3047*, 2009.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Roger Koenker and Kevin F. Hallock. Quantile regression. *Journal of Economic Perspectives*, 15(4):143–156, December 2001. doi: 10.1257/jep.15.4.143. URL `https://www.aeaweb.org/articles?id=10.1257/jep.15.4.143`.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, Univ. Toronto, 2009.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22 (1):79 – 86, 1951. doi: 10.1214/aoms/1177729694. URL `https://doi.org/10.1214/aoms/1177729694`.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

Eduardo Ley. On the peculiar distribution of the us stock indexes' digits. *The American Statistician*, 50(4): 311–313, 1996.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Simon Newcomb. Note on the frequency of use of different digits in natural numbers. *American Journal of Mathematics*, 4:39–40, 1881.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pp. 625–632, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102430. URL `https://doi.org/10.1145/1102351.1102430`.

Laura O'Mahony, David JP O'Sullivan, and Nikola S Nikolov. On the detection of anomalous or out-of-distribution data in vision models using statistical techniques. In *The 3rd International Conference on Artificial Intelligence and Computer Vision (AICV2023), March 5–7, 2023*, pp. 426–435. Springer, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *ArXiv*, abs/1912.01703, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Surya Kant Sahu, Abhinav Java, and Arshad Shaikh. Rethinking neural networks with benford's law. In *Workshop at the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 12 2021.

Lijing Shao and Bo-Qiang Ma. The significant digit law in statistical physics. *Physica A: Statistical Mechanics and its Applications*, 389(16):3109–3116, 2010.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Luohan Wang and Bo-Qiang Ma. A concise proof of benford's law. *Fundamental Research*, 2023. ISSN 2667-3258. doi: https://doi.org/10.1016/j.fmre.2023.01.002.

P. Warden. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *ArXiv e-prints*, April 2018. URL `https://arxiv.org/abs/1804.03209`.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

A Appendix

You may include other additional sections here.