Disparity Between Batches as a Signal for Early Stopping

Mahsa Forouzesh [✉] and Patrick Thiran

Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
{mahsa.forouzesh, patrick.thiran}@epfl.ch

Abstract. We propose a metric for evaluating the generalization ability of deep neural networks trained with mini-batch gradient descent. Our metric, called *gradient disparity*, is the ℓ_2 norm distance between the gradient vectors of two mini-batches drawn from the training set. It is derived from a probabilistic upper bound on the difference between the classification errors over a given mini-batch, when the network is trained on this mini-batch and when the network is trained on another mini-batch of points sampled from the same dataset. We empirically show that gradient disparity is a very promising early-stopping criterion (i) when data is limited, as it uses all the samples for training and (ii) when available data has noisy labels, as it signals overfitting better than the validation data. Furthermore, we show in a wide range of experimental settings that gradient disparity is strongly related to the generalization error between the training and test sets, and that it is also very informative about the level of label noise.

Keywords: Early Stopping · Generalization · Gradient Alignment · Overfitting · Neural Networks · Limited Datasets · Noisy Labels.

1 Introduction

Early-stopping using a separate validation set is one of the most popular techniques used to avoid under/over fitting deep neural networks trained with iterative methods, such as gradient descent [1–3]. The optimization is stopped when the performance of the model on a validation set starts to diverge from its performance on the training set. Early stopping requires an accurately labeled validation set, separated from the training set, to act as an unbiased proxy on the unseen test error. Obtaining such a reliable validation set can be expensive in many real-world applications as data collection is a time-consuming process that might require domain expertise. Furthermore, deep learning is becoming popular in applications for which there is simply not enough available data [4, 5]. Finally, inexperienced label collectors, complex tasks (e.g., distinguishing a guinea pig from a hamster), and corrupted labels due for instance to adversarial attacks result in datasets that contain noisy labels [6]. Deep neural networks have the unfortunate ability to overfit to such small and/or noisy labeled datasets, an issue that cannot be completely solved by popular regularization techniques [7].

A signal of overfitting during training is therefore particularly useful, if it does *not* need a separate, accurately labeled validation set, which is the purpose of this paper.

Let S_1 and S_2 be two mini-batches of points sampled from the available (training) dataset. Suppose that S_1 is selected for an iteration (step) of the mini-batch gradient descent (SGD), at the end of which the parameter vector is updated to w_1. The average loss over S_1 (denoted by $L_{S_1}(h_{w_1})$) is in principle reduced, given a sufficiently small learning rate. The average loss $L_{S_2}(h_{w_1})$ over the other mini-batch S_2 is not as likely to be reduced. It is more likely to remain larger than the loss $L_{S_2}(h_{w_2})$ computed over S_2, if it was S_2 instead of S_1 that had been selected for this iteration. The difference $\mathcal{R}_2 = L_{S_2}(h_{w_1}) - L_{S_2}(h_{w_2})$ is the penalty that we pay for choosing S_1 over S_2 (and similarly, $\mathcal{R}_1$ is the penalty that we would pay for choosing S_2 over S_1). $\mathcal{R}_2$ is illustrated in Fig. 1 for a hypothetical non-convex loss as a function of a one dimensional parameter w. The expected penalty measures how much, in an iteration, a model updated on one batch (S_1) is able to generalize on average to another batch (S_2) from the dataset. Hence, we call $\mathcal{R}$ the *generalization penalty*.

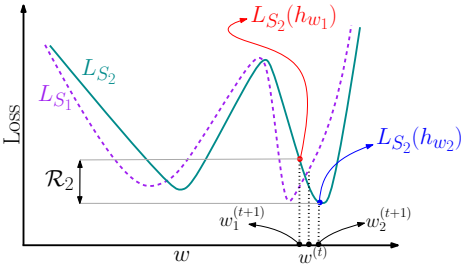

Fig. 1. An illustration of the penalty term $\mathcal{R}_2$, where the y-axis is the loss, and the x-axis indicates the parameter of the model. L_{S_1} and L_{S_2} are the average losses over mini-batches S_1 and S_2, respectively. $w^{(t)}$ is the parameter at iteration t and $w_i^{(t+1)}$ is the parameter at iteration $t+1$ if batch S_i was selected for the update step at iteration t, with $i \in \{1, 2\}$.

We establish a probabilistic upper bound on the sum of the expected penalties $\mathbb{E}[\mathcal{R}_1] + \mathbb{E}[\mathcal{R}_2]$ by adapting the PAC-Bayesian framework [8–10], given a pair of mini-batches S_1 and S_2 sampled from the dataset (Theorem 1). Interestingly, under some mild assumptions, this upper bound is essentially a simple expression driven by $\|g_1 - g_2\|_2$, where g_1 and g_2 are the gradient vectors over the two mini-batches S_1 and S_2, respectively. We call it *gradient disparity*: it measures how much a small gradient step on one mini-batch negatively affects the performance on the other one.

We propose gradient disparity as an effective early stopping criterion, because of its computational tractability that makes it simple to use during the course of training, and because of its strong link with generalization error, as evidenced in the experiments that we run on state-of-the-art configurations. Gradient disparity is particularly well suited when the available dataset has limited labeled data, because it does not require splitting the available dataset into training and validation sets: all the available data can be used during training, unlike for instance k-fold cross-validation. We observe that using gradient disparity, instead of an unbiased validation set, results in a predictive improvement of at least

Table 1. The test loss and area under the receiver operating characteristic curve (AUC score) of the MRNet dataset [11] when using 5-fold cross-validation (5-fold CV) and gradient disparity (GD) as early stopping criteria for detecting the presence of abnormally, ACL tears, and meniscal tears from the sagittal plane MRI scans. The corresponding curves during training are shown in Fig. 9 (see Appendix F.3 for more details). The results of early stopping are given, both when the metric (GD or validation loss) has increased for 5 epochs from the beginning of training and between parenthesis when the metric has increased for 5 consecutive epochs. Using GD outperforms 5-fold CV with either choice of the early stopping threshold. The standard deviations are obtained from 5 runs.

Task	Method	Test Loss	Test AUC Score (in percentage)
Abnormal	5-fold CV	$0.284_{\pm 0.016}(0.307_{\pm 0.057})$	$71.016_{\pm 3.66}(87.44_{\pm 1.35})$
	GD	$\mathbf{0.274}_{\pm 0.004}(\mathbf{0.275}_{\pm 0.053})$	$\mathbf{72.67}_{\pm 3.85}(\mathbf{88.12}_{\pm 0.35})$
ACL	5-fold CV	$0.973_{\pm 0.111}(1.246_{\pm 0.142})$	$79.80_{\pm 1.23}(89.32_{\pm 1.47})$
	GD	$\mathbf{0.842}_{\pm 0.101}(\mathbf{1.136}_{\pm 0.121})$	$\mathbf{81.81}_{\pm 1.64}(\mathbf{91.52}_{\pm 0.09})$
Meniscal	5-fold CV	$0.758_{\pm 0.04}(1.163_{\pm 0.127})$	$73.53_{\pm 1.30}(72.14_{\pm 0.74})$
	GD	$\mathbf{0.726}_{\pm 0.019}(\mathbf{1.14}_{\pm 0.323})$	$\mathbf{74.08}_{\pm 0.79}(\mathbf{73.80}_{\pm 0.24})$

1% for classification tasks with limited and very costly available data, such as the MRNet dataset, which is a small size image-classification dataset used for detecting knee injuries (Table 1).

Moreover, we find that gradient disparity is a more accurate early stopping criterion than validation loss when the available dataset contains noisy labels. Gradient disparity reflects the label noise level quite well throughout the training process, especially at early stages of training. Finally, we observe that gradient disparity has a strong positive correlation with the test error across experimental settings that differ in training set size, batch size, and network width.

2 Related Work

The coherent gradient hypothesis [12] states that the gradient is stronger in directions where similar examples exist and towards which the parameter update is biased. He and Su [13] study the local elasticity phenomenon, which measures how the prediction over one sample changes, as the network is updated on another sample. Motivated by [13], reference [14] proposes generalization upper bounds using locally elastic stability. The generalization penalty introduced in our work measures how the prediction over one sample (batch) changes when the network is updated on the same sample, instead of being updated on another sample.

Finding a practical metric that completely captures the generalization properties of deep neural networks, and in particular indicates the level of label noise and decreases with the size of the training set, is still an active research direction [15–18]. Recently, there have been a few studies that propose similarity between gradients as a generalization metric. The benefit of tracking generalization by measuring the similarity between gradient vectors is its tractability

during training, and the dispensable access to unseen data. Sankararaman et al. [19] propose gradient confusion, which is a bound on the inner product of two gradient vectors, and shows that the larger the gradient confusion is, the slower the convergence is. Gradient interference (when the gradient inner product is negative) has been studied in multi-task learning, reinforcement learning and temporal difference learning [20–22]. Yin et al. [23] study the relation between gradient diversity, which measures the dissimilarity between gradient vectors, and the convergence performance of distributed SGD algorithms. Fort et al. [24] propose a metric called stiffness, which is the cosine similarity between two gradient vectors, and shows empirically that it is related to generalization. Fu et al. [25] study the cosine similarity between two gradient vectors for natural language processing tasks. Reference [26] measures the alignment between the gradient vectors within the same class (denoted by Ω_c) , and studies the relation between Ω_c and generalization as the scale of initialization (the variance of the probability distribution the network parameters are initially drawn from) is increased. These metrics are usually not meant to be used as early stopping criteria, and indeed in Table 2 and Table 12 in the appendix, we observe that none of them consistently outperforms k-fold cross-validation.

Another interesting line of work is the study of the variance of gradients in deep learning settings. Negrea et al. [27] derive mutual information generalization error bounds for stochastic gradient Langevin dynamics (SGLD) as a function of the sum (over the iterations) of square gradient incoherences, which is closely related to the variance of gradients. Two-sample gradient incoherences also appear in [28], which are taken between a training sample and a "ghost" sample that is not used during training and therefore taken from a validation set (unlike gradient disparity). The upper bounds in [27, 28] are cumulative bounds that increase with the number of iterations and are not intended to be used as early stopping criteria. As shown in Appendix H, gradient disparity can be used as an early stopping criterion not only for SGD with additive noise (such as SGLD), but also other adaptive optimizers. Reference [29] shows that the variance of gradients is a decreasing function of the batch size. However, reference [30] hypothesizes that gradient variance counter-intuitively increases with the batch size, by studying the effect of the learning rate on the variance of gradients, which is consistent with our results on convolutional neural networks in Section 6. References [29, 30] mention the connection between variance of gradients and generalization as promising future directions. Our study shows that variance of gradients used as an early stopping criterion outperforms k-fold cross-validation (see Table 12).

Liu et al. [31] propose a relation between gradient signal-to-noise ratio (SNR), called GSNR, and the one-step generalization error, with the assumption that both the training and test sets are large. Mahsereci et al. [32] also study gradient SNR and propose an early stopping criterion called evidence-based criterion (EB) that eliminates the need for a held-out validation set. Reference [33] proposes an early stopping criterion based on the signal-to-noise ratio figure, which is further studied in [34], a study that shows the average test error achieved by standard early stopping is lower than the one obtained by this criterion. Zhang et al. [35]

Table 2. The test error (TE) and test loss (TL) achieved by using various metrics as early stopping criteria for an AlexNet trained on the MNIST dataset with 50% random labels. See Table 12 in the appendix for further details and experiments.

	Min	GD/Var	EB	GSNR	$g_i \cdot g_j$	sign$(g_i \cdot g_j)$	cos$(g_i \cdot g_j)$	Ω_c	OV	k-fold	No ES
TE	13.76	**16.66**	24.63	35.68	37.92	24.63	35.68	29.40	34.36	17.86	25.72
TL	0.75	1.08	**0.86**	1.68	1.82	**0.86**	1.68	1.46	1.65	1.09	0.91

empirically show that the variance term in the bias-variance decomposition of the loss function dominates the variations of the test loss, and hence propose optimization variance (OV) as an early stopping criterion.

Summary of Comparison to Related Work In Table 2 and Appendix I, we compare gradient disparity (GD) to EB, GSNR, gradient inner product, sign of the gradient inner product, variance of gradients, cosine similarity, Ω_c, and OV. We observe that the only metrics that consistently outperform k-fold cross-validation as early stopping criteria across various settings (see Table 12 in the appendix), and that reflect well the label noise level (see in Figs. 22 and 23 that metrics such as EB and sign$(g_i \cdot g_j)$ do not correctly detect the label noise level), are gradient disparity and variance of gradients. The two are analytically very close as discussed in Appendix I.2. However, we observe that the correlation between gradient disparity and the test loss is in general larger than the correlation between variance of gradients and the test loss (see Table 13 in the appendix).

3 Generalization Penalty

Consider a classification task with input $x \in \mathcal{X} \coloneqq \mathbb{R}^n$ and ground truth label $y \in \{1, 2, \cdots, k\}$, where k is the number of classes. Let $h_w \in \mathcal{H} : \mathcal{X} \to \mathcal{Y} \coloneqq \mathbb{R}^k$ be a predictor (classifier) parameterized by the parameter vector $w \in \mathbb{R}^d$, and $l(\cdot, \cdot)$ be the 0-1 loss function $l\left(h_w(x), y\right) = \mathbb{1}\left[h_w(x)[y] < \max_{j \neq y} h_w(x)[j]\right]$ for all $h_w \in \mathcal{H}$ and $(x, y) \in \mathcal{X} \times \{1, 2, \cdots, k\}$. The expected loss and the empirical loss over the training set S of size m are respectively defined as

$$L(h_w) = \mathbb{E}_{(x,y)\sim D}\left[l\left(h_w(x), y\right)\right], \tag{1}$$

and

$$L_S(h_w) = \frac{1}{m}\sum_{i=1}^{m} l(h_w(x_i), y_i), \tag{2}$$

where D is the probability distribution of the data points and $S = \{(x_i, y_i)\}^m$ is a collection of m i.i.d. samples drawn from D. Similar to the notation used in [15], distributions on the hypotheses space $\mathcal{H}$ are simply distributions on the underlying parameterization. With some abuse of notation, ∇L_{S_i} refers to the

gradient with respect to the surrogate differentiable loss function, which in our experiments is cross entropy[1].

In a mini-batch gradient descent (SGD) setting, let mini-batches S_1 and S_2 have sizes m_1 and m_2, respectively, with $m_1 + m_2 \leq m$. Let $w = w^{(t)}$ be the parameter vector at the beginning of an iteration t. If S_1 is selected for the next iteration, w gets updated to $w_1 = w^{(t+1)}$ with

$$w_1 = w - \gamma \nabla L_{S_1}(h_w), \tag{3}$$

where γ is the learning rate. The generalization penalty $\mathcal{R}_2$ is defined as the gap between the loss over S_2, $L_{S_2}(h_{w_1})$, and its target value, $L_{S_2}(h_{w_2})$, at the end of iteration t.

When selecting S_1 for the parameter update, Eq. (3) makes a step towards learning the input-output relations of mini-batch S_1. If this negatively affects the performance on mini-batch S_2, $\mathcal{R}_2$ will be large; the model is learning the data structures that are unique to S_1 and that do not appear in S_2. Because S_1 and S_2 are mini-batches of points sampled from the same distribution D, they have data structures in common. If, throughout the learning process, we consistently observe that, in each update step, the model learns structures unique to only one mini-batch, then it is very likely that the model is memorizing the labels instead of learning the common data-structures. This is captured by the generalization penalty $\mathcal{R}$.

We adapt the PAC-Bayesian framework [8, 9] to account for the trajectory of the learning algorithm; For each learning iteration t we define a prior, and two possible posteriors depending on the choice of the mini-batch selection. Let $w \sim P$ follow a prior distribution P, which is a $\mathcal{F}_t$-measurable function, where $\mathcal{F}_t$ denotes the filtration of the available information at the beginning of iteration t. Let h_{w_1}, h_{w_2} be the two learned single predictors, at the end of iteration t, from S_1 and S_2, respectively. In this framework, for $i \in \{1, 2\}$, each predictor h_{w_i} is randomized and becomes h_{ν_i} with $\nu_i = w_i + u_i$, where u_i is a random variable whose distribution might depend on S_i. Let Q_i be the distribution of ν_i, which is a distribution over the predictor space $\mathcal{H}$ that depends on S_i via w_i and possibly u_i. Let $\mathcal{G}_i$ be a σ-field such that $\sigma(S_i) \cup \mathcal{F}_t \subset \mathcal{G}_i$ and such that the posterior distribution Q_i is $\mathcal{G}_i$-measurable for $i \in \{1, 2\}$. We further assume that the random variable $\nu_1 \sim Q_1$ is statistically independent from the draw of the mini-batch S_2 and, vice versa, that $\nu_2 \sim Q_2$ is independent from the batch S_1[2], i.e., $\mathcal{G}_1 \perp\!\!\!\perp \sigma(S_2)$ and $\mathcal{G}_2 \perp\!\!\!\perp \sigma(S_1)$.

Theorem 1. *For any $\delta \in (0, 1]$, with probability at least $1 - \delta$ over the sampling of sets S_1 and S_2, the sum of the expected penalties conditional on S_1, and S_2,*

[1] We have also studied networks trained with the mean square error in Appendix E.3, and we observe that there is a strong positive correlation between the test error/loss and gradient disparity for this choice of the surrogate loss function as well (see Fig. 5).

[2] Mini-batches S_1 and S_2 are drawn without replacement, and the random selection of indices of mini-batches S_1 and S_2 is independent from the dataset S. Hence, similarly to [27, 36], we have $\sigma(S_1) \perp\!\!\!\perp \sigma(S_2)$.

respectively, satisfies

$$\mathbb{E}\left[\mathcal{R}_1\right] + \mathbb{E}\left[\mathcal{R}_2\right] \leq \sqrt{\frac{2KL(Q_2||Q_1) + 2\ln\frac{2m_2}{\delta}}{m_2 - 2}} + \sqrt{\frac{2KL(Q_1||Q_2) + 2\ln\frac{2m_1}{\delta}}{m_1 - 2}}. \tag{4}$$

In this paper, the goal is to get a signal of overfitting that indicates at the beginning of each iteration t whether to stop or to continue training. This signal should track the performance of the model at the end of iteration t by investigating its evolution over all the possible outcomes of the batch sampling process during this iteration. For simplicity, we consider two possible outcomes: either mini-batch S_1 or mini-batch S_2 is chosen for this iteration (we later in the next section extend to more pairs of mini-batches). If we were to use bounds such as the ones in [10, 37] for one iteration at a time, the generalization error at the end of that iteration can be bounded by a function of either $\mathrm{KL}(Q_1||P)$ or $\mathrm{KL}(Q_2||P)$, depending on the selected mini-batch. Therefore, as each of the two mini-batches is equally likely to be sampled, we should track $\mathrm{KL}(Q_1||P)$ and $\mathrm{KL}(Q_2||P)$ for a signal of overfitting at the end of the iteration, which requires in turn access to the three distributions P, Q_1 and Q_2. In contrast, the upper bound on the generalization penalty given in Theorem 1 only requires the two distributions Q_1 and Q_2, which is a first step towards a simpler metric since, loosely speaking, the symmetry between the random choices for S_1 and S_2 should carry over these two distributions, leading us to assume the random perturbations u_1 and u_2 to be identically distributed. If furthermore we assume them to be Gaussian, then we show in the next section that $\mathrm{KL}(Q_2||Q_1)$ and $\mathrm{KL}(Q_1||Q_2)$ are equal and boil down to a very tractable generalization metric, which we call gradient disparity.

4 Gradient Disparity

In Section 3, the randomness modeled by the additional perturbation u_i, conditioned on the current mini-batch S_i, comes from (i) the parameter vector at the beginning of the iteration w, which itself comes from the random parameter initialization and the stochasticity of the parameter updates until that iteration, and (ii) the gradient vector ∇L_{S_i} (simply denoted by g_i), which may also be random because of the possible additional randomness in the network structure due for instance to dropout [38]. A common assumption made in the literature is that the random perturbation u_i follows a normal distribution [37, 39]. The upper bound in Theorem 1 takes a particularly simple form if we assume that for $i \in \{1, 2\}$, u_i are zero mean i.i.d. normal variables ($u_i \sim \mathcal{N}(0, \sigma^2 I)$), and that w_i is fixed, as in the setting of [15].

As $w_i = w - \gamma g_i$ for $i \in \{1, 2\}$, the KL-divergence between $Q_1 = \mathcal{N}(w_1, \sigma^2 I)$ and $Q_2 = \mathcal{N}(w_2, \sigma^2 I)$ (Lemma 1 in Appendix B) is simply

$$\mathrm{KL}(Q_1||Q_2) = \frac{1}{2}\frac{\gamma^2}{\sigma^2}\left\|g_1 - g_2\right\|_2^2 = \mathrm{KL}(Q_2||Q_1), \tag{5}$$

which shows that, keeping a constant step size γ and assuming the same variance for the random perturbations σ^2 in all the steps of the training, the bound in Theorem 1 is driven by $\|g_1 - g_2\|_2$. This indicates that the smaller the ℓ_2 distance between gradient vectors is, the lower the upper bound on the generalization penalty is, and therefore the closer the performance of a model trained on one mini-batch is to a model trained on another mini-batch.

For two mini-batches of points S_i and S_j, with respective gradient vectors g_i and g_j, we define the *gradient disparity* (GD) between S_i and S_j as

$$\mathcal{D}_{i,j} = \|g_i - g_j\|_2 \,. \tag{6}$$

To compute $\mathcal{D}_{i,j}$, a first option is to sample S_i from the training set and S_j from the held-out validation set, which we refer to as the "train-val" setting, following [24]. The generalization penalty $\mathcal{R}_j$ in this setting measures how much, during the course of an iteration, a model updated on a training set is able to generalize to a validation set, making the resulting ("train-val") gradient disparity $\mathcal{D}_{i,j}$ a natural candidate for tracking overfitting. But it requires access to a validation set to sample S_j, which we want to avoid. The second option is to sample both S_i and S_j from the training set, as proposed in this paper, to yield now a value of $\mathcal{D}_{i,j}$ that we could call "train-train" gradient disparity (GD) by analogy. Importantly, we observe a strong positive correlation between the two types of gradient disparities ($\rho = 0.957$) in Fig. 2. Therefore, we can expect that both of them do (almost) equally well in detecting overfitting, with the advantage that the latter does not require to set data aside, contrary to the former. We will therefore consider GD when both batches are sampled from the training set and evaluate it in this paper.

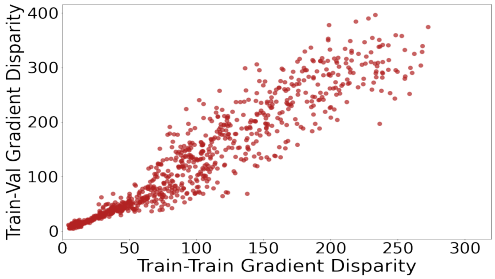

Fig. 2. "Train-val" gradient disparity versus "train-train" gradient disparity for 220 experimental settings that vary in architecture, dataset, training set size, label noise level and initial random seed. Pearson's correlation coefficient is $\rho = 0.957$.

To track the upper bound of the generalization penalty for more pairs of batches, we can compute an average gradient disparity over B batches, which requires all the B gradient vectors at each iteration, which is computationally

expensive if B is large. We approximate it by computing GD over only a much smaller subset of the batches, of size $s \ll B$,

$$\overline{\mathcal{D}} = \sum_{i=1}^{s} \sum_{j=1, j\neq i}^{s} \frac{\mathcal{D}_{i,j}}{s(s-1)}.$$

In our experiments, $s = 5$; we observe that such a small subset is already sufficient (see Appendix E.2 for an experimental comparison of different values of s).

Consider two training iterations t_1 and t_2 where $t_1 \ll t_2$. At earlier stages of the training (iteration t_1), the parameter vector ($w^{(t_1)}$) is likely to be located in a steep region of the training loss landscape, where the gradient vector of training batches, g_i, and the training loss $L_{S_i}(h_{w^{(t_1)}})$ take large values. At later stages of training (iteration t_2), the parameter vector ($w^{(t_2)}$) is more likely in a flatter region of the training loss landscape where g_i and $L_{S_i}(h_{w^{(t_2)}})$ take small values. To compensate for this scale mismatch when comparing the distance between gradient vectors at different stages of training, we re-scale the loss values within each batch before computing $\overline{\mathcal{D}}$ (see Appendix E.1 for more details). Note that this re-scaling is only done for the purpose of using GD as a metric, and therefore does not have any effect on the training process itself.

We focus on the vanilla SGD optimizer. In Appendix H, we extend the analysis to other stochastic optimization algorithms: SGD with momentum, Adagrad, Adadelta, and Adam. In all these optimizers, we observe that GD (Eq. (6)) appears in $\mathrm{KL}(Q_1||Q_2)$ with other factors that depend on a decaying average of past gradient vectors. Experimental results support the use of GD as an early stopping metric also for these popular optimizers (see Fig. 21 in Appendix H). For vanilla SGD optimizer, we also provide an alternative and simpler derivation leading to gradient disparity from the linearization of the loss function in Appendix D.

5 Early Stopping Criterion

In the presence of *large* amounts of *reliable* data, it is affordable to split the available dataset into a training and a validation set and to perform early stopping by evaluating the performance of the model on the held-out validation set. However, if the dataset is *limited*, this approach makes an inefficient use of the data because the model never learns the information that is still present in the validation set. Moreover, if the dataset is *noisy*, held-out validation might poorly estimate the performance of the model as the validation set might contain a high percentage of noisy samples. To avoid these issues, k-fold cross-validation [40] is a solution that makes an efficient usage of the available data while providing an unbiased estimate of the performance, at the expense of a high computational overhead and of a possibly underestimated variance [41]. While each of its k rounds is itself a setting with a held-out validation set, k-fold cross-validation (as opposed to held-out validation) would be therefore advantageous to use in the presence of limited and/or noisy data. It extracts more information from the

Table 3. The test loss and accuracy when using gradient disparity (GD) and k-fold cross-validation (CV) (k =5) as early stopping criteria when the available dataset is limited: (top) VGG-13 trained on 1.28 k samples of the CIFAR-10 dataset, and (bottom) AlexNet trained on 256 samples of the MNIST dataset. The corresponding curves during training are presented in Fig. 7. The results below are obtained by stopping the optimization when the metric (either validation loss or GD) has increased for 5 epochs from the beginning of training.

Setting	Method	Test loss	Test accuracy
CIFAR-10, VGG-13	5-fold CV	$1.846_{\pm 0.016}$	$35.982_{\pm 0.393}$
	GD	$\mathbf{1.793}_{\pm 0.016}$	$\mathbf{36.96}_{\pm 0.861}$
MNIST, AlexNet	5-fold CV	$1.123_{\pm 0.25}$	$62.62_{\pm 6.36}$
	GD	$\mathbf{0.656}_{\pm 0.080}$	$\mathbf{79.12}_{\pm 3.04}$

dataset as it uses all the data samples for both training and validation, and it is less dependent on how the data is split into training and validation sets.

The baseline to beat is therefore k-fold cross-validation (CV). We compare gradient disparity to CV in the two target settings: (i) when the available dataset is limited and (ii) when the available dataset has corrupted labels. Medical applications are one of the practical examples of setting (i), where datasets are costly because they require the collection of patient data, and the medical staff's expertise to label the data. An example of such an application is the MRNet dataset [11], which contains a limited number of MRI scans to study the presence of abnormally, ACL tears and meniscal tears in knee injuries. This dataset is by nature limited and we use the entire available data for both early stopping methods GD and k-fold CV. In addition, to further simulate setting (i), we use small subsets of three image classification benchmark datasets: MNIST, CIFAR-10 and CIFAR-100. Performing early stopping in the presence of label noise (setting (ii)) is also practically very important, because it has been empirically observed that deep neural networks trained on noisy datasets overfit to noisy labeled samples at later stages of training. A good early stopping signal can therefore prevent such an overfitting [42–44]. To simulate setting (ii), we use a corrupted version of these image classification benchmark datasets, where for a fraction of the samples (the amount of noise), we choose the labels at random.

(i) We observe that using gradient disparity instead of a validation loss in k-fold CV results in an improvement of more than 1% (on average over all three tasks) in the test AUC score of the MRNet dataset, and therefore adds a correct detection for more than one patient for each task (see Table 1). Furthermore, we observe that gradient disparity performs better than k-fold CV as an early stopping criterion for image-classification benchmark datasets as well (see Table 3). A plausible explanation for the better peformance of GD over k-fold CV is that, although CV uses the entire set of samples over the k rounds for both training and validation, it trains the model only on a $(1 - 1/k)$ portion of the dataset in each individual round. In contrast, GD allows to train the model over the entire

dataset in a single run, which therefore results in a better performance on the final unseen (test) data when data is limited. For more experimental results refer to Table 10 and Figs. 7 and 9 in Appendix F.

(ii) We observe that gradient disparity performs better than k-fold cross-validation as an early stopping criterion when data is noisy (see Table 4). When the labels of the available data are noisy, the validation set is no longer a reliable estimate of the test set. Nevertheless, and although it is computed over the noisy training set, gradient disparity reflects the performance on the test set quite well[3] For more experimental results refer to Table 11 and Fig. 8 in Appendix F.

Quite surprisingly, we observe that GD performs better in terms of accuracy than an extension of k-fold CV, which we call k^+-fold CV, which uses the entire dataset for training with the early stopping signal found by k-fold CV (see Table 4, where $k = 10$ for these settings). More precisely, k^+-fold CV is done in 3 steps: (1) perform k-fold CV, (2) compute the stopping epoch by tracking the validation loss found in step (1), and (3) retrain the model on the entire dataset and stop at the epoch obtained in step (2). k^+-fold CV uses therefore $k + 1$ rounds because of step (3), thus one more round than k-fold CV, but unlike k-fold CV (and similarly to GD), k^+-fold CV produces models that are trained on the entire dataset. It is therefore interesting to note that using GD still outperforms k^+-fold CV in terms of accuracy (although not in terms of loss).

Table 4. The test loss and accuracy when using gradient disparity (GD) and k-fold cross-validation (CV) (k =10) as early stopping criteria when the available dataset is noisy: 50% of the available data has random labels. The corresponding curves during training are shown in Fig. 8. The results below are obtained by stopping the optimization when the metric (either validation loss or GD) has increased for 5 epochs from the beginning of training. The last row in each setting, which we call 10^+-fold CV, refers to the test loss and accuracy reached at the epoch suggested by 10-fold CV, for a network trained on the entire set. Notice that for the CIFAR-100 experiments (the top rows), for computational reasons, the models are trained on only 1.28 k samples of the dataset which explains the very low test accuracy for this experiment. However, for the MNIST experiments (the bottom rows), the models are trained on the entire dataset, and we observe rather high test accuracies.

Setting	Method	Test loss	Test accuracy
CIFAR-100, ResNet-18	10-fold CV	$5.023_{\pm 0.083}$	$1.59_{\pm 0.15}$ (top-5: $6.47_{\pm 0.52}$)
	GD	$\mathbf{4.463}_{\pm 0.038}$	$\mathbf{3.68}_{\pm 0.52}$ (top-5: $\mathbf{15.22}_{\pm 1.24}$)
	10^+-fold CV	$4.964_{\pm 0.057}$	$1.68_{\pm 0.24}$ (top-5: $7.05_{\pm 0.71}$)
MNIST, AlexNet	10-fold CV	$0.656_{\pm 0.034}$	$97.28_{\pm 0.20}$
	GD	$0.654_{\pm 0.031}$	$\mathbf{97.32}_{\pm 0.27}$
	10^+-fold CV	$\mathbf{0.639}_{\pm 0.029}$	$97.31_{\pm 0.15}$

[3] See for example Fig. 8 (left column) where the validation loss fails to estimate the test loss, but where GD (Fig. 8 (middle left column)) does signal overfitting correctly.

The metrics used as early stopping criteria, whether they are the validation loss or gradient disparity, are measured on signals that are subject to random fluctuations. As a result, they rely on a pre-defined threshold p (sometimes called *patience* by practitioners) that sets the number of iterations during which the metric increases before the algorithm is stopped. We use two popular thresholds: (t1) the first one is to stop the algorithm when the metric (GD or validation loss) has increased for $p = 5$ (possibly non consecutive) epochs from the beginning of training, and (t2) the second is the same as (t1) but the $p = 5$ epochs must be consecutive. Both GD and k-fold CV might be sensitive to the choice of (t1) or (t2), or even to the value of p itself. It is therefore important to study the sensitivity of an early stopping metric to the choice of the threshold p, which is done in Appendix F.1 for both GD and k-fold CV for ten different values of $p \in \{1, \cdots, 10\}$ and the two thresholds (t1) and (t2). We observe that GD always gives similar or higher test accuracy than k-fold CV for all 20 possible thresholds (see Fig. 6). More importantly, GD is much more robust to the choice of the early stopping threshold (see Table 5).

Table 5. Sensitivity of each method to the choice of the early stopping threshold. The sensitivity is computed from the reported values of Tables 7 and 8 according to Eq. 14 in the appendix.

Method	Sensitivity of the Test Accuracy	Sensitivity of the Test Loss
GD	**0.916**	**0.886**
CV	1.613	1.019

When data is abundant and clean, the validation loss is affordable and trustworthy to use as an early stopping signal. GD does also correctly signal overfitting in this case (see for example Fig. 13 in the appendix). However, when data is limited and/or noisy (which is also when early stopping is particularly important), we observe that the validation loss is costly and unreliable to use as an early stopping signal. In contrast, in these settings, GD does not cost a separate held-out validation set and is a reliable signal of overfitting even in the presence of label noise. In practice, the label noise level of a given dataset is in general not known a priori and we do not know whether the size of the dataset is large enough to afford sacrificing a subset for validation. We often do not know whether we are in the former setting, with abundant and clean data, or in the later setting, with limited and/or noisy data. It is therefore important to have a good early stopping criterion that works for both settings. Unlike the validation loss, GD is such a signal.

6 Discussion and Final Remarks

We propose gradient disparity (GD), as a simple to compute early stopping criterion that is particularly well-suited when the dataset is limited and/or noisy. Beyond indicating the early stopping time, GD is well aligned with factors that contribute to improve or degrade the generalization performance of a model, which have an often strikingly similar effect on the value of GD as well. We briefly discuss in this section some of these observations that further validate the use of GD as an effective early stopping criterion; more details are provided in the appendix.

Label Noise Level. We observe that GD reflects well the label noise level throughout the training process, even at early stages of training, where the generalization gap fails to do so (see Figs. 10, 12, 16, and 20 in Appendix D).

Training Set Size. We observe that GD, similarly to the test error, decreases with training set size, unlike many previous metrics as shown by [16, 17]. Moreover, we observe that applying data augmentation decreases the values of both GD and the test error (see Figs. 17 and 18 in Appendix G).

Batch Size. We observe that both the test error and GD increase with batch size. This observation is counter-intuitive because one might expect that gradient vectors get more similar when they are averaged over a larger batch. GD matches the ranking of test errors for different networks, trained with different batch sizes, as long as the batch sizes are not too large (see Fig. 19 in Appendix G).

Width. We observe that both the test error and GD (normalized with respect to the number of parameters) decrease with the network width for ResNet, VGG and fully connected neural networks (see Fig. 15 in Appendix G).

Gradient disparity belongs to the same class of metrics based on the similarity between two gradient vectors [19, 24–26, 30]. A common drawback of all these metrics is that they are not informative when the gradient vectors are very small. In practice however, we observe (see for instance Figure 13 in the appendix) that the time at which the test and training losses start to diverge, which is the time when overfitting kicks in, does not only coincide with the time at which gradient disparity increases, but also occurs much before the training loss becomes infinitesimal. This drawback is therefore unlikely to cause a problem for gradient disparity when it is used as an early stopping criterion. Nevertheless, as a future direction, it would be interesting to explore this further especially for scenarios such as epoch-wise double-descent [45].

References

1. Prechelt, L.: Early stopping-but when? In: Neural Networks: Tricks of the trade, pp. 55–69. Springer (1998)
2. Yao, Y., Rosasco, L., Caponnetto, A.: On early stopping in gradient descent learning. Constructive Approximation 26(2), 289–315 (2007)
3. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al.: Recent advances in convolutional neural networks. Pattern Recognition 77, 354–377 (2018)

4. Roh, Y., Heo, G., Whang, S.E.: A survey on data collection for machine learning: a big data-ai integration perspective. IEEE Transactions on Knowledge and Data Engineering (2019)
5. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on amazon mechanical turk. In: Proceedings of the ACM SIGKDD workshop on human computation. pp. 64–67 (2010)
6. Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE transactions on neural networks and learning systems 25(5), 845–869 (2013)
7. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530 (2016)
8. McAllester, D.A.: Pac-bayesian model averaging. In: Proceedings of the twelfth annual conference on Computational learning theory. pp. 164–170 (1999)
9. McAllester, D.A.: Some pac-bayesian theorems. Machine Learning 37(3), 355–363 (1999)
10. McAllester, D.: Simplified pac-bayesian margin bounds. In: Learning theory and Kernel machines, pp. 203–215. Springer (2003)
11. Bien, N., Rajpurkar, P., Ball, R.L., Irvin, J., Park, A., Jones, E., Bereket, M., Patel, B.N., Yeom, K.W., Shpanskaya, K., et al.: Deep-learning-assisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of mrnet. PLoS medicine 15(11), e1002699 (2018)
12. Chatterjee, S.: Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=ryeFY0EFwS
13. He, H., Su, W.: The local elasticity of neural networks. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=HJxMYANtPH
14. Deng, Z., He, H., Su, W.J.: Toward better generalization bounds with locally elastic stability. arXiv preprint arXiv:2010.13988 (2020)
15. Dziugaite, G.K., Roy, D.M.: Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. arXiv preprint arXiv:1703.11008 (2017)
16. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. In: Advances in Neural Information Processing Systems. pp. 5947–5956 (2017)
17. Nagarajan, V., Kolter, J.Z.: Uniform convergence may be unable to explain generalization in deep learning. In: Advances in Neural Information Processing Systems. pp. 11611–11622 (2019)
18. Chatterji, N., Neyshabur, B., Sedghi, H.: The intriguing role of module criticality in the generalization of deep networks. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=S1e4jkSKvB
19. Sankararaman, K.A., De, S., Xu, Z., Huang, W.R., Goldstein, T.: The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. arXiv preprint arXiv:1904.06963 (2019)
20. Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., Tesauro, G.: Learning to learn without forgetting by maximizing transfer and minimizing interference. arXiv preprint arXiv:1810.11910 (2018)
21. Liu, V., Yao, H., White, M.: Toward understanding catastrophic interference in value-based reinforcement learning. Optimization Foundations for Reinforcement Learning Workshop at NeurIPS (2019)
22. Bengio, E., Pineau, J., Precup, D.: Interference and generalization in temporal difference learning. arXiv preprint arXiv:2003.06350 (2020)

23. Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K., Bartlett, P.: Gradient diversity: a key ingredient for scalable distributed learning. arXiv preprint arXiv:1706.05699 (2017)
24. Fort, S., Nowak, P.K., Jastrzebski, S., Narayanan, S.: Stiffness: A new perspective on generalization in neural networks. arXiv preprint arXiv:1901.09491 (2019)
25. Fu, J., Liu, P., Zhang, Q., Huang, X.: Rethinking generalization of neural models: A named entity recognition case study. arXiv preprint arXiv:2001.03844 (2020)
26. Mehta, H., Cutkosky, A., Neyshabur, B.: Extreme memorization via scale of initialization. arXiv preprint arXiv:2008.13363 (2020)
27. Negrea, J., Haghifam, M., Dziugaite, G.K., Khisti, A., Roy, D.M.: Information-theoretic generalization bounds for sgld via data-dependent estimates. In: Advances in Neural Information Processing Systems. pp. 11013–11023 (2019)
28. Haghifam, M., Negrea, J., Khisti, A., Roy, D.M., Dziugaite, G.K.: Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms. arXiv preprint arXiv:2004.12983 (2020)
29. Qian, X., Klabjan, D.: The impact of the mini-batch size on the variance of gradients in stochastic gradient descent. arXiv preprint arXiv:2004.13146 (2020)
30. Jastrzebski, S., Szymczak, M., Fort, S., Arpit, D., Tabor, J., Cho, K., Geras, K.: The break-even point on optimization trajectories of deep neural networks. arXiv preprint arXiv:2002.09572 (2020)
31. Liu, J., Bai, Y., Jiang, G., Chen, T., Wang, H.: Understanding why neural networks generalize well through gsnr of parameters. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=HyevIJStwH
32. Mahsereci, M., Balles, L., Lassner, C., Hennig, P.: Early stopping without a validation set. arXiv preprint arXiv:1703.09580 (2017)
33. Liu, Y., Starzyk, J.A., Zhu, Z.: Optimized approximation algorithm in neural networks without overfitting. IEEE transactions on neural networks 19(6), 983–995 (2008)
34. Piotrowski, A.P., Napiorkowski, J.J.: A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling. Journal of Hydrology 476, 97–111 (2013)
35. Zhang, X., Wu, D., Xiong, H., Dai, B.: Optimization variance: Exploring generalization properties of {dnn}s (2021), https://openreview.net/forum?id=ZAfeFYKUek5
36. Dziugaite, G.K., Hsu, K., Gharbieh, W., Roy, D.M.: On the role of data in pac-bayes bounds. arXiv preprint arXiv:2006.10929 (2020)
37. Neyshabur, B., Bhojanapalli, S., Srebro, N.: A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. arXiv preprint arXiv:1707.09564 (2017)
38. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15(1), 1929–1958 (2014)
39. Bellido, I., Fiesler, E.: Do backpropagation trained neural networks have normal weight distributions? In: International Conference on Artificial Neural Networks. pp. 772–775. Springer (1993)
40. Stone, M.: Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society: Series B (Methodological) 36(2), 111–133 (1974)
41. Bengio, Y., Grandvalet, Y.: No unbiased estimator of the variance of k-fold cross-validation. Journal of machine learning research 5(Sep), 1089–1105 (2004)

42. Li, M., Soltanolkotabi, M., Oymak, S.: Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In: International Conference on Artificial Intelligence and Statistics. pp. 4313–4324. PMLR (2020)
43. Song, H., Kim, M., Park, D., Lee, J.G.: Prestopping: How does early stopping help generalization against label noise? (2020), https://openreview.net/forum?id=BklSwn4tDH
44. Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., Chang, Y.: Robust early-learning: Hindering the memorization of noisy labels. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=Eql5b1_hTE4
45. Heckel, R., Yilmaz, F.F.: Early stopping in deep networks: Double descent and how to eliminate it. arXiv preprint arXiv:2007.10099 (2020)