Are All Training Examples Created Equal? An Empirical Study

Kailas Vodrahalli
UC Berkeley
kailasv@berkeley.edu

Ke Li
UC Berkeley
ke.li@eecs.berkeley.edu

Jitendra Malik
UC Berkeley
malik@eecs.berkeley.edu

Abstract

Modern computer vision algorithms often rely on very large training datasets. However, it is conceivable that a carefully selected subsample of the dataset is sufficient for training. In this paper, we propose a gradient-based importance measure that we use to empirically analyze relative importance of training images in four datasets of varying complexity. We find that in some cases, a small subsample is indeed sufficient for training. For other datasets, however, the relative differences in importance are negligible. These results have important implications for active learning on deep networks. Additionally, our analysis method can be used as a general tool to better understand diversity of training examples in datasets.

1. Introduction

Deep learning has achieved remarkable success in recent years, largely made possible by more powerful hardware and an abundance of training data. In particular, computer vision has been enabled by deep convolutional neural networks to attain near or exceeding human performance on a variety of tasks including image classification [6], object segmentation [2], pose recognition [14], image synthesis [8], and many others. These deep networks are typically trained using stochastic gradient methods where data is subsampled in minibatches and the network parameters are updated by the gradient of the parameter weights relative to a loss function for the given minibatch (modulated by some learning rate and other hyperparameters).

As these deep networks typically have millions of learnable parameters, they require millions of images for training. In many cases, large datasets are not publically available and are expensive to collect and annotate. So, a common procedure is to pretrain a network using a large image dataset and subsequently finetune the network for a specific application using a smaller, application-specific dataset.

Despite the importance of these datasets for many applications, there is a dearth of analysis on their properties. In this paper, we seek to remedy this through an analysis over several well-known datasets: MNIST [12], CIFAR-10 [10], CIFAR-100 [10], and ImageNet [3]. In particular, we are interested in understanding the relative importance of images for training a deep neural network and determining how diverse these datasets are. For the purposes of this paper, we restrict ourselves to image datasets and select image classification as our application. It is, however, feasible to apply our techniques to other datasets and tasks. We conduct this analysis by computing the gradient magnitude of the loss corresponding to each individual training image at the end of training to determine a relative importance score for each image. We then retrain our network on subsets of the data selected based on our importance measure to determine how well these subsets capture the distribution of the entire dataset.

We propose two attributes to compare datasets: simplicity and redundancy. Using these attributes, we conclude that while some common datasets like CIFAR-10/100 and ImageNet do indeed contain diverse images, others like MNIST have much redundancy in data. Our analysis has implications for the difficulty of performing active learning in the context of deep learning and for dataset collection practices.

Our paper is organized as follows. In section 2 we outline related works that motivate our analysis technique and past results on dataset analysis. In section 3 we describe our methods. Section 4 contains our results on four standard image datasets, and section 5 provides additional discussion and potential implications of and extensions to our work. We conclude in section 6.

2. Related Work

In this paper, we are interested in understanding the relative importance of image data in common datasets. This concept is related to two areas of research: active learning and data distribution analysis. In active learning, the goal is to adaptively subsample data to reduce the number of observations required to train a model. This sampling selects the most informative data (which can change over the course of training). Similarly, in analyzing the distribution of data, we are interested in understanding the similarity of datapoints and how the data is distributed. This concept is relevant for

active learning, where the most informative examples can be better identified if a data distribution is known.

2.1. Active Learning

Active learning is a machine learning paradigm that uses an oracle to interactively query data on which to train a model. The oracle chooses data to help the model learn, allowing for faster convergence and less data usage overall. An excellent survey on the field can be found in [19].

This approach is useful in several contexts. For situations where unlabeled data is abundant but labeling is expensive, an active learning approach can help select a limited subset of the data that needs to be labeled to train a model adequately. Active learning can also help decrease training time by effectively decreasing dataset size.

Empirical results have demonstrated success for active learning [19]. However, there is evidence that it may not always work, and there are situations where random sampling performs better than some active learning algorithms [17, 5]. These results suggest that active learning may be dataset, model, and application dependent, at least to a certain extent.

One model where active learning has seen success is support vector machines (SVM). Here, active learning has been applied to such tasks as image and text classification [22, 13] among others. Intuitively, active learning makes sense in the SVM case, as in practice the decision boundaries in SVMs often depend on only a few points in a dataset. Recent work suggests that active learning may be possible in the deep neural network setting [18, 4, 7].

Our analysis approach is based on ideas in active learning and takes additional inspiration from the notion of importance given to support vectors in an SVM. In particular, we take the individual images that induce the largest gradient magnitudes to be our 'support vectors' in the sense that they are the most critical to training. Given the SVM intuition, we do not actively query this data, but rather fix a large batch of data at the start.

2.2. Data Distribution Analysis

Typical machine learning datasets are published with several surface-level properties. In computer vision tasks, this may include such information as image labels, image sizes, and image acquisition method (e.g., synthetically generated, crowd sourced, or curated). Depending on task, the image labels may include, for example, one or multiple class names (e.g., for classification), one or multiple object bounding boxes (e.g., for object detection), or an annotated figure (e.g., for pose recognition). Some datasets also contain additional annotation levels. For example, ImageNet contains a semantic tree that describes relations between categories. Additional distribution-level properties like mean pixel value or pixel value variance are easily computed. However, these properties do not fully model relation of the dataset to the true data manifold they describe.

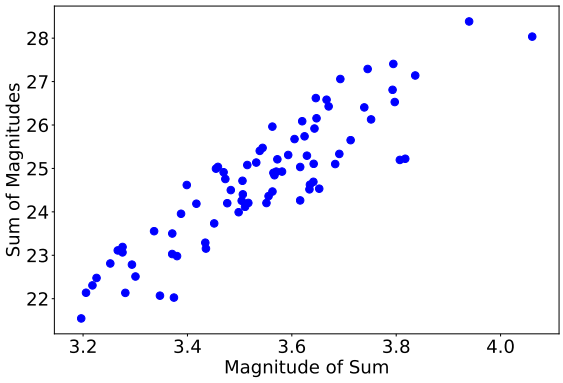


Figure 1: Comparison of $||\sum_{i=1}^{N} \nabla_\theta L_{i,\theta}||$ and $\sum_{i=1}^{N} ||\nabla_\theta L_{i,\theta}||$ (see (6)) for randomly sampled mini-batches from ImageNet using the VGG16 model. Mini-batch size is determined by the size used during training. Linear correlations suggest that the sum of gradient magnitudes is a reasonable approximation for ordering gradients.

There has been some relatively recent work such as [23], which analyze inherent bias in datasets. This bias is in relation to the world-view of the true data, and it reflects various biases in a specific dataset (e.g., in an image dataset containing 'car' images, this may show up as the angles at which a car is photographed, or the typical model of car photographed). Various other work support these results and offer solutions to debias the data like weighting points according to dataset or other image properties [9, 15]. Some datasets also have more nefarious forms of biases due to limited diversity of geographical origin [20] or due to the underlying data having undesirable bias as in the corpus of English text [1].

In this paper, we seek to extend knowledge on the intrinsic properties of datasets by analyzing the relative importance of data in large image datasets. In particular, we are interested in the diversity / redundancy of data. Previous methods to measure diversity look at basic image-level statistics [3] or various human-annotated qualities like amount of texture or distinctiveness [16]. We believe it will be informative to investigate diversity from a trained model's perspective. Given the almost universal practice of using convolutional neural networks, we choose these networks as the models we investigate.

3. Methods

In Section 4, we analyze four standard image datasets: MNIST, CIFAR-10, CIFAR-100, and ImageNet. For each

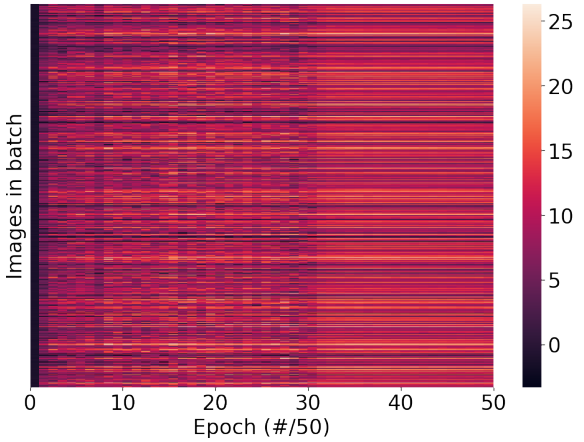


Figure 2: Heatmap of gradient magnitudes for 512 MNIST images over 50 epochs of training. Black corresponds to large and white corresponds to small gradient magnitude. Color is on a log scale. Abrupt change at 30 occurs due to learning rate decay.

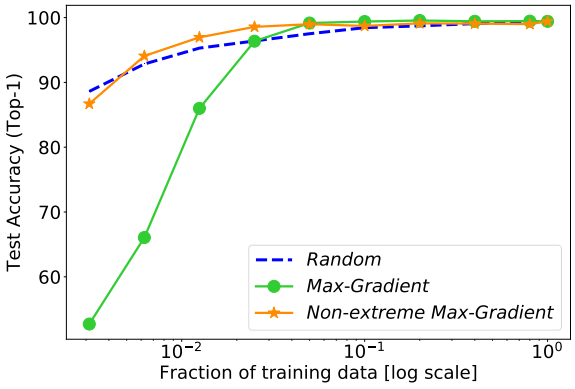


Figure 3: Top-1 test accuracy for MNIST. *Non-extreme Max-Gradient* overtakes *Random* when using 0.6% of training data. *Max-Gradient* overtakes *Random* when using 3% of training data.

dataset we select between one and three standard network architectures to perform our analysis. We also randomly subsample 10% of the data for validation data prior to any analysis. Here we describe our analysis method.

3.1. Gradient Magnitude as Importance Measure

In our analysis, we need a measure of importance in order to subsample our set of training images. For this purpose, we will use the gradients of terms in the loss corresponding to individual training images w.r.t. the parameters of a fully trained deep network. Here we justify why this approach is reasonable.

In most cases, the training objective can be written in the following form:

$$f_\theta^* = \arg\min_{f_\theta \in \mathcal{F}_\theta} \mathcal{L}(f_\theta) \tag{1}$$

where

$$\mathcal{L}(f_\theta) = \left(\frac{1}{N}\sum_{i=1}^{N} l(f_\theta(\mathbf{x_i}), y_i)\right) + \mathcal{R}(f_\theta) \tag{2}$$

$$= \left(\frac{1}{N}\sum_{i=1}^{N} L_{i,\theta}\right) + \mathcal{R}(f_\theta). \tag{3}$$

Here, $\mathbf{x_i}$ is the i^{th} image and y_i is its corresponding label, N is the total number of images in the training set, l is defined as the cross-entropy loss, $\mathcal{R}$ is some form of weight regularization (we omit the associated hyperparameters inside the function for clarity), and f_θ is our neural network (interpret as $\mathcal{F}_\theta$ is the set of possible neural networks given our standard architecture, and θ defines a specific set of parameters over that architecture).

So, the loss is a sum of losses over the training images. As the gradient is a linear operator, the gradient of the loss will be the sum of gradients over individual images. As is standard, we use a variant of stochastic gradient descent (SGD) to perform this optimization. Letting θ_t be our provisional network parameters after t training iterations, η_t be our learning rate at iteration t, and $\mathcal{B}_t$ be our minibatch at iteration t, we have

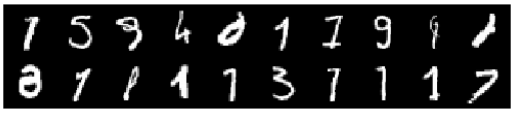
$$\theta_{t+1} \leftarrow \theta_t - \left(\frac{\eta_t}{|\mathcal{B}_t|}\sum_{i\in\mathcal{B}_t} \nabla_\theta L_{i,\theta_t}\right) - \eta_t \nabla_\theta \mathcal{R}(f_{\theta_t}) \tag{4}$$

We consider the magnitude of change in parameters from one iteration to the next:

$$\|\theta_{t+1} - \theta_t\| = \left\| \left(\frac{\eta_t}{|\mathcal{B}_t|}\sum_{i\in\mathcal{B}_t} \nabla_\theta L_{i,\theta_t}\right) - \eta_t \nabla_\theta \mathcal{R}(f_{\theta_t}) \right\| \tag{5}$$

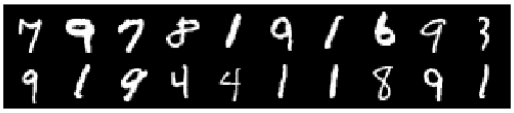
The magnitude of change in the parameters is directly related to how important/informative the current batch of training data is. If the current batch is important, then the model should change significantly after seeing the current batch. On the other hand, if it is not important, the model should remain almost the same.

This observation doesn't directly tell us which individual training example is important. To find the important training examples, we upper bound the magnitude of change in the parameters. This is a conservative estimate of the importance of the batch, in the sense that a batch that is important is guaranteed to achieve a high upper bound, though there could be false positives, where a batch that is not important also achieves a high upper bound. This upper bound is obtained by applying the triangle inequality:

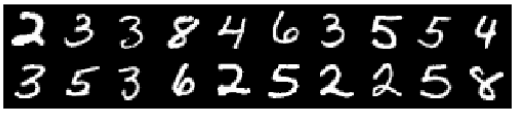


(a) 99^{th} percentile

(b) 95^{th} percentile

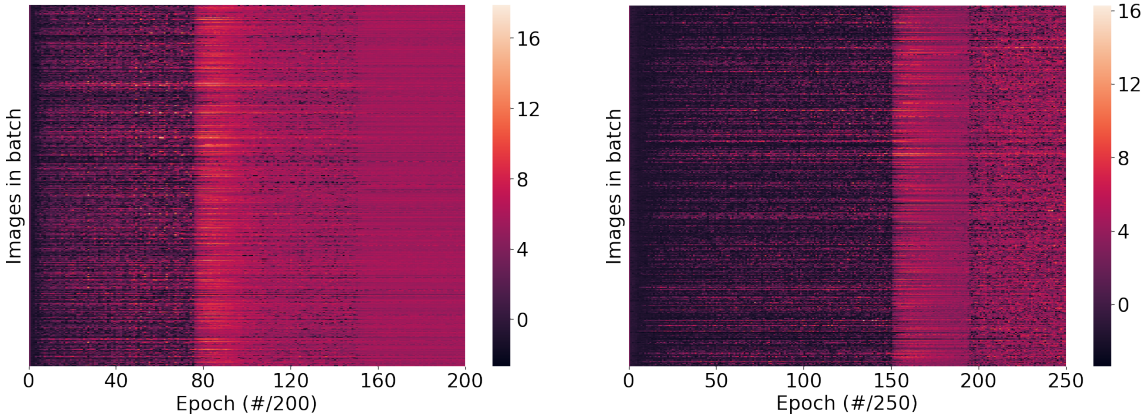


(c) 50^{th} percentile



(d) 5^{th} percentile

Figure 4: Sample images from MNIST organized by their final gradient magnitude.



(a) CIFAR-10 for VGG16 model; 200 epochs.

(b) CIFAR-100 for VGG16 model; 250 epochs.

Figure 5: Heatmap of gradient magnitudes for 512 CIFAR-10/100 images. Black corresponds to large and white corresponds to small gradient magnitude. Color is on a log scale. Abrupt changes due to learning rate decay and likely saddle points.

$$\left\| \sum_{i \in \mathcal{B}_t} \nabla_\theta L_{i,\theta} \right\| \leq \sum_{i \in \mathcal{B}_t} \|\nabla_\theta L_{i,\theta}\| . \tag{6}$$

We can then find which training example contributes the most to this upper bound by selecting the examples that have the largest gradient magnitudes $\|\nabla_\theta L_{i,\theta}\|$. One possible concern is that some training examples that are selected may contribute a lot to the upper bound, but may not contribute significantly to the original quantity. However, this is justifiable because it is better to err on the conservative side and detect all examples that *could* be important and cause large changes to our parameter values. One mitigating factor is that our models are highly expressive and can have millions of parameters. Because two gradient vectors are likely to be nearly orthogonal in high dimensions, the bound is unlikely to be very loose, since that would require the gradients of some training examples to be nearly colinear.

To confirm this intuition empirically, in Figure 1, we plot randomly sampled minibatches from ImageNet showing the original quantity, $||\sum_{i=1}^{N} \nabla_\theta L_{i,\theta}||$ and the upper bound, $\sum_{i=1}^{N} ||\nabla_\theta L_{i,\theta}||$. The linear correlation between the quantities suggests that the upper bound is a good approximation for ordering gradient values, as larger individual gradient magnitudes tend to correspond to larger overall gradients for the batch.

Then we can select a size k subsample, $\mathcal{B}$, based on this upper bound:

$$\mathcal{B}^* = \max_{\mathcal{B}:|\mathcal{B}|=k} \sum_{i=1}^{N} ||\nabla_\theta L_{i,\theta}||. \tag{7}$$

This set $\mathcal{B}^*$ is the set we choose by selecting the top-k images with largest gradient magnitude.

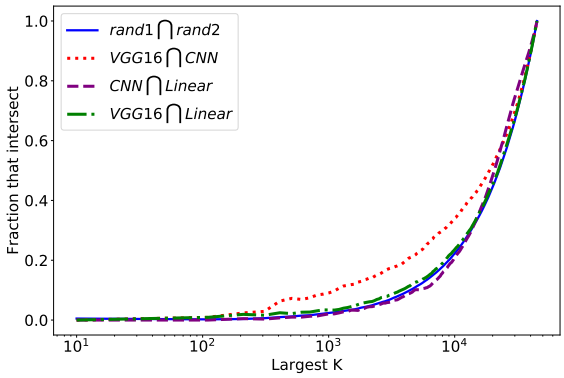


Figure 6: This plot compares the similarity of the gradient-based ordering that different models impose on the CIFAR-10 training images. Let $S_{i,k}$ be the size-k set of images with largest gradient magnitudes for model i. Then each plotted line computes $|S_{i_1,k} \bigcap S_{i_2,k}|/k$, as a function of k. $rand1 \bigcap rand2$ is a baseline that computes the size of intersection for two random size k subsets. Interestingly, the size of the intersection between VGG16 and the generic CNN is significant (notice we plot with log-scaled X-axis).

Algorithm 1 Gradient Analysis

procedure ANALYSIS(f_θ)
  Train network f_θ on all data
  Compute test accuracy
  for $i = 1, ..., N$ **do**
  return SUBSAMPLE_ANALYSIS(f_θ)

procedure SUBSAMPLE_ANALYSIS(f_θ)
  Subsample data using ∇_i ▷ See Section 3.3
  Retrain network f_θ on subsampled data only
  return data subsample, test accuracy

3.2. Subsample Selection

Our analysis procedure for each dataset is described in Algorithm 1. The main procedure, ANALYSIS, can be broken into three steps:

(2-3) Train the network using the entirety of the training data, using validation data for early stopping. Log the test accuracy.

(4-5) Compute the gradient of each network parameter with respect to a loss for each training image in the training dataset. We will use these gradients to subsample data in the next step.

(6) Now retrain the network from a random initialization using a subsampled portion of the data. Then log the test accuracy as a measure of how well the subset represents the entirety of the dataset. We further describe how we subsample data in Section 3.3.

3.3. Batch Selection

We propose three methods of sampling data based on gradients. We also include a random subsample baseline.

a. *Random*: This is our baseline approach. We randomly select the given number of images from all training images.

b. *Max-Gradient*: We select images in descending order by their gradient magnitude until we reach the given number of images.

c. *Non-extreme Max-Gradient*: We order images by their gradient magnitude in descending order. Then we discard the top 5% of images, and proceed to select images in order until we reach the given number of images.

d. *Gradient-CDF*: Here, 'CDF' stands for 'cumulative distribution function.' We use the gradient magnitudes to induce a probability mass function (PMF) over the training images: Letting g_i denote the gradient magnitude for the i^{th} image, we define the PMF at $\mathbf{x_i}$ as

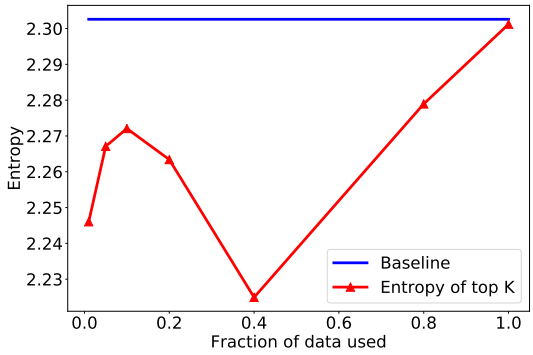
$$P(\mathbf{x_i}) = \frac{g_i}{\sum_{i=1}^{N} g_i} \tag{8}$$

We subsequently use the resulting distribution to sample, without replacement, the given number of images.

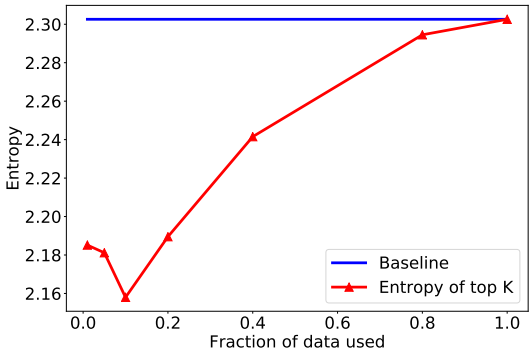
Max-Gradient is our original approach; the intuition is based in selecting images that have the largest affect on network weights through SGD. We find, however, that there are correlations between gradients that cause this method to artificially skew the distribution of images (see Figure 7). Additionally, we observe that the images with largest gradients may just be outliers or poor examples (see Figure 4). So we propose *Non-extreme Max-Gradient* as an alternative that can decrease the number of outliers and increase diversity of images, and *Gradient-CDF* as an alternative that softens how we select images by adding randomness into the process.

4. Experiments

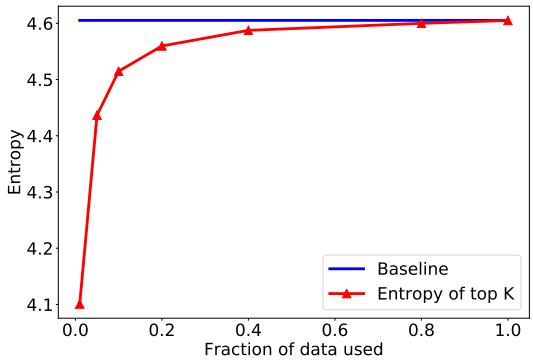
We apply our analysis technique to four well-known image datasets: MNIST, CIFAR-10, CIFAR-100, and ImageNet. To compute the gradient magnitudes used in our subsampling procedure, there are several options. We considered the following norms to apply to the gradient vector: ℓ_1, ℓ_2, and ℓ_∞ norms. We also considered computing the norm over subsets of all parameters. In particular, (A) biases only, (B) weights only, and (C) last layer weights only



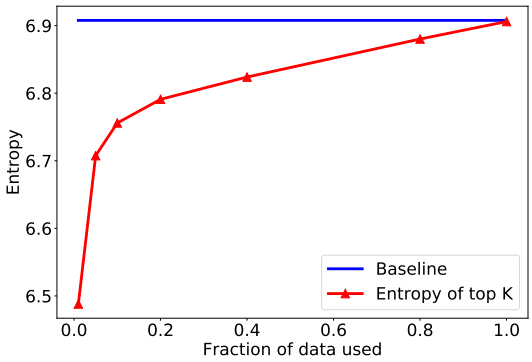
(a) MNIST for (generic) CNN model



(b) CIFAR-10 for VGG16 model

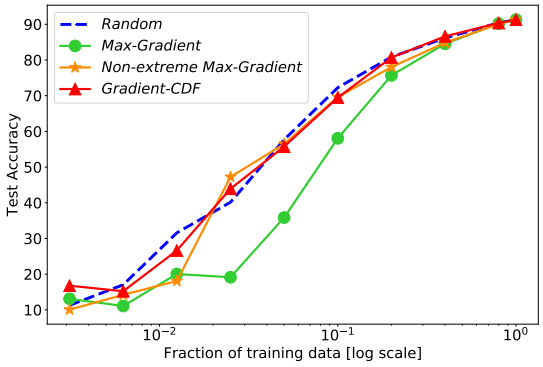


(c) CIFAR-100 for VGG16 model

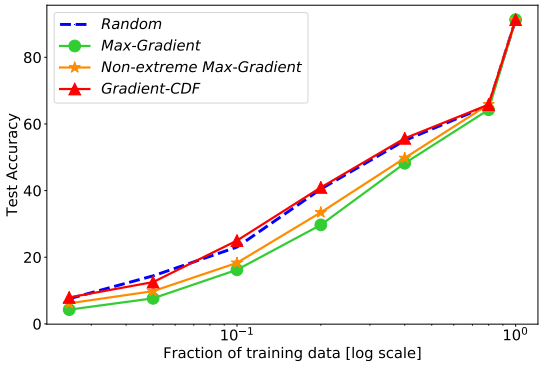


(d) ImageNet for VGG16 model

Figure 7: Entropy of class labels for top-k images ordered by gradient magnitude on the given dataset and trained model. Baseline refers to the entropy for uniformly balanced label frequencies.



(a) CIFAR-10 for VGG16 model

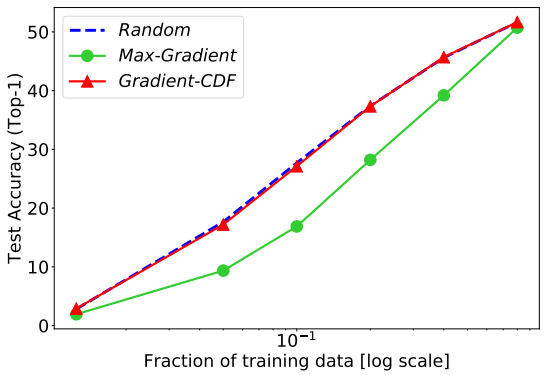


(b) CIFAR-100 for VGG16 model

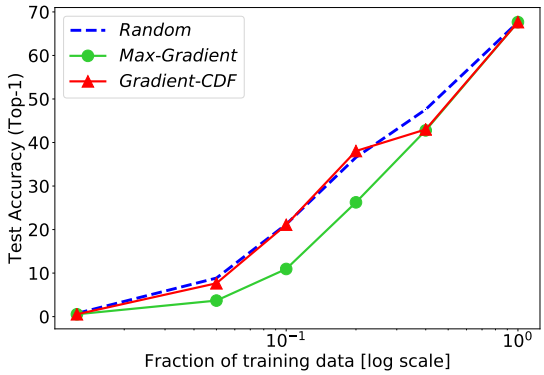
Figure 8: Top-1 test accuracy for CIFAR-10/100 run on a VGG16 network.

(this may be reasonable as the magnitude of the gradient in each network layer can vary significantly). We observed little difference between these choices on the MNIST and CIFAR-10 datasets. Given computational limitations, we only report results using ℓ_2 norm and option (C) for MNIST, CIFAR-10, and CIFAR-100; we use the ℓ_2 norm and option (B) for ImageNet to provide some variety.

We consider the following standard networks: VGG16 [21] and AlexNet [11]. We use a few additional shallow networks for comparisons on MNIST and CIFAR-10.



(a) AlexNet; last data point uses 80% of the dataset



(b) VGG16; last data point computed with pretrained model

Figure 9: Top-1 test accuracy for ImageNet. Gradients used from sampling computed using a VGG16 network.

4.1. MNIST

For this study, we consider a network architecture that consists of 2 convolution layers followed by two fully connected layers. Given that classification on MNIST is generally considered easy, we did not tune this architecture. Indeed, this model attains 99.42% test accuracy. .

Before performing the main analysis, it is useful to understand the relative magnitudes of gradients at the end of training. In Figure 2, we plot a heatmap showing the negative log magnitude of gradients for 512 randomly sampled data points throughout training. At the beginning of training, all of these gradient magnitudes start roughly equal. Over time, all magnitudes decrease. By the end of training some images have much smaller gradients than others. So, effectively, towards the end of training, only a subset of training examples contribute significantly to the parameter updates. Intuitively, these correspond to the hard/more important training examples.

Now we present the result of our gradient analysis, subsampling data using several methods. In Figure 3, we plot the final top-1 test accuracy for *Random*, *Max-Gradient*, and *Non-extreme Max-Gradient* subsampling. Notice that for very small batch sizes (corresponding to 175 total training images), random subsampling outperforms either gradient based approaches. Additionally, *Non-extreme Max-Gradient* outperforms the regular *Max-Gradient* approach. One possible explanation is that the top few images with largest gradients could be outliers and may not be representative of the dataset.

However, when more data is used, *Max-Gradient* outperforms the other two methods. This result is especially interesting as it suggests we need some 'easy' images to be able to train the neural net successfully, but the hardest images (with the largest gradient magnitudes) are still the most important. As this leader change happens at only 5% of the training data, and moreover because the test accuracy of *Max-Gradient* is already nearly the test accuracy when using all the data, this suggests that MNIST has much redundant data and can be well approximated by a substantially condensed subset. The fact that this subset can be found by examining the magnitudes of gradients suggest that gradients are a reasonable measure of importance for each data example.

It is also informative to visualize the training images with the largest gradient. Looking at Figure 4, it is clear that images with largest gradient are difficult to classify, while the rest get progressively easier to read.

4.2. CIFAR-10

We analyze CIFAR-10 using three network architectures to show how various models behave and to demonstrate how our analysis can apply in general. The architectures are: (1) a linear classifier, (2) and five layer CNN, and (3) VGG16 adapted for CIFAR-10.

In Figure 6, we compare how the set of largest gradient images vary across network architectures. VGG16 and the CNN consistently have a higher overlap than the random baseline. However, this overlap is still a small percentage of the subsampled dataset size for small subsample amounts, suggesting that the images with largest gradient are somewhat specific to a given model. An interesting intuition here is that the most difficult images in a dataset can be model-specific rather than intrinsic to the image.

To determine whether gradients are correlated, we can compute the entropy of the class labels for the top-k images. In Figure 7, we plot this entropy for different values of k. It can be seen that the entropy is lower when we select less data, confirming the existence of correlations between largest gradient and class label. Comparing the plots for MNIST and CIFAR-10, we see that there is greater skew in distribution for CIFAR-10, perhaps contributing to the

decrease in test accuracy from *Max-Gradient* subsampling. We take this as a motivation for using the *Gradient-CDF* method.

Now we look at some comparisons on the VGG16 models. In Figure 5a we show how gradients for a randomly selected batch of training images change over training. The two abrupt changes result from learning rate decay. Also note that the gradients are colored on a log scale, so there are significant differences in gradient value at the end of training. However, compared to MNIST, the differences are less striking and suggest that the importance of images in CIFAR-10 is more distributed. Figure 8a compares the test accuracy of the final model trained using various subsampling procedures. We can see that gradient-based sampling can give an increase in test accuracy, but random sampling can be better. This is in stark contrast to our results on MNIST where gradient-based sampling was clearly beneficial.

4.3. CIFAR-100

CIFAR-100 is similar to CIFAR-10, but has 100 rather than 10 classes. We briefly report results of our analysis here. We use the VGG16 network adapted to the CIFAR-100 image size.

Looking at the heatmap in Figure 5b, it does seem like there are large variations in gradient magnitude at the end of training. However, the test accuracy plot in Figure 8b clearly suggests that gradient-based sampling performs worse than random sampling. Figure 7 shows that the largest gradients have a significantly skewed label distribution, potentially suggesting that certain classes are more difficult to classify than other classes.

4.4. ImageNet

The ImageNet dataset consists of 1000 diverse classes and over a million images of varying sizes. As is standard, we scale all images down to 256x256 and take 224x224 croppings to train our network. We consider two network architectures: AlexNet and VGG16. For both architectures, we sample data using the gradient information from only the VGG16 architecture, giving us insight in how well the gradient-based importance measure generalizes between networks. Given the results from CIFAR-100 and CIFAR-10, we only include analysis for the following sampling methods: *Random*, *Max-Gradient*, and *Gradient-CDF*.

Because the test set of ImageNet is not publicly available, we use the official validation set as our test set (and, as is true for all our datasets, we subsample our train set to attain a train and validation set).

For AlexNet, the results we see are similar to those from CIFAR-100. In particular, Figure 9a shows that the *Max-Gradient* sampling method result in lower performance than either *Random* or *Gradient-CDF* sampling. For VGG16 we see very similar trends in Figure 9b until 40% of data is sampled. This result is especially interesting as it suggests that the gradient-based importance values may be similar between these two deep networks.

5. Discussion

The results we have presented provide several insights. On MNIST, subsampling by maximum gradient gives higher performance, suggesting that there is indeed redundancy in the dataset. However, when we move to CIFAR-10 and CIFAR-100, random sampling performs better than *Max-Gradient* and is most closely matched by *Gradient-CDF*, which is simply weighted random sampling. One indicator for why this may be the case is in the heatmaps shown in Figure 5. At the end of training, the gradient magnitudes are much closer together than they are in other datasets, perhaps suggesting that all images have roughly the same 'difficulty' as seen by the network. So, there may not be a small subset of data that captures the entirety of either CIFAR dataset; in other words, CIFAR seems to be diverse in the sense that it is not redundant.

We also note that gradient-based sampling may not always be optimal. Looking at Figure 7, it is apparent that sampling by gradient skews the class distribution when we order by gradient magnitude, which in turn makes generalization more difficult. Note that in CIFAR-100, *Non-extreme Max-Gradient* results in lower test accuracy, while in CIFAR-10 it achieved roughly the same performance as *Random*. This difference may be due to CIFAR-100 having 10 times as many classes, and so the issue of image distribution skew is exacerbated.

We see the same behavior for ImageNet. In Figures 9b and 9a, *Gradient-CDF* closely matches *Random* in test accuracy, while *Max-Gradient* achieves a significantly lower test accuracy. In Figure 7, we see a similar shape as in CIFAR-100. However, interestingly, it appears that the top gradient magnitude images for ImageNet are more varied than for CIFAR-100. This may be due to an even greater diversity of images in ImageNet than in CIFAR-100.

These observations suggest two takeaways:

(1) MNIST contains redundant data. CIFAR-10, CIFAR-100, and ImageNet contain little redundant data. This conclusion is especially interesting considering that the properties of CIFAR-10 and MNIST are similar in some ways: both contain the same number of classes, and number of bytes in CIFAR-10 images is only roughly 4x the number in MNIST images (i.e., the data size is on the same scale). There are two potential reasons: (1) the dataset itself is collected in such a way that the images are more diverse, and (2) the underlying space of dogs, cats, and other animals and vehi-

cles is inherently larger than that for Arabic numerals, despite the image data representation size being relatively close for both datasets. While intuitively obvious to some degree, it is interesting to see explicit evidence supporting this hypothesis.

(2) It might be significantly more difficult to reduce the amount of training data while maintaining performance on CIFAR-10, CIFAR-100, and ImageNet than on MNIST, suggesting that active learning on CIFAR-10, CIFAR-100, and ImageNet might be substantially more challenging than on MNIST.

6. Conclusion

Given the importance of large datasets in modern machine learning, it is critical to understand dataset properties in order to better exploit the data. In this paper, we have proposed an approach to empirically analyze the diversity of data and applied this method on four image datasets of varying complexity. For CIFAR-10, CIFAR-100, and ImageNet, we have found that most training examples are valuable, while for MNIST we found that most training examples are largely redundant. The results also suggest the large number of images in CIFAR-10, CIFAR-100, and ImageNet are indeed necessary, and it is difficult to reduce the amount of data while maintaining performance.

The results from our empirical analysis are specific to the deep learning models used in obtaining our results rather than a direct property of the dataset. Interestingly, however, we have found evidence that the analysis results do generalize between models suggesting that some images are important, independent of the learning model.

While we selected gradient magnitudes as a method for analyzing importance, it would be interesting in the future to look into other importance metrics. In particular, given that gradients are indeed correlated, it may be worthwhile to consider importance metrics that are evaluated on batches rather than individual images. This idea has been successfully utilized in the active learning setting [5, 24]. With this framework, it would be possible to enforce properties like approximate orthogonality of gradients inside a batch and high entropy of class labels, both of which are seemingly important for high performance on small datasets. However, there is a tradeoff in complexity with the possibility of exponential computational complexity (in the size of the subset). So, heuristics or approximate optimization approaches may be necessary.

References

[1] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357, 2016. 2

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 1

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 1, 2

[4] M. Ducoffe and F. Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018. 2

[5] Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In *Advances in neural information processing systems*, pages 593–600, 2008. 2, 9

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 1

[7] S.-J. Huang, J.-W. Zhao, and Z.-Y. Liu. Cost-effective training of deep cnns with active model adaptation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 1580–1588, New York, NY, USA, 2018. ACM. 2

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[9] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012. 2

[10] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 1

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 6

[12] Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 1

[13] X. Li, L. Wang, and E. Sung. Multilabel svm active learning for image classification. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 4, pages 2207–2210. IEEE, 2004. 2

[14] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. 1

[15] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. 2

[16] O. Russakovsky, J. Deng, Z. Huang, A. C. Berg, and L. Fei-Fei. Detecting avocados to zucchinis: what have we done, and where are we going? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2064–2071, 2013. 2

[17] A. I. Schein and L. H. Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007. 2

[18] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. 2

[19] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012. 2

[20] S. Shankar, Y. Halpern, E. Breck, J. Atwood, J. Wilson, and D. Sculley. No classification without representation: Assessing geodiversity issues in open data sets for the developing world. In *NIPS 2017 workshop: Machine Learning for the Developing World*, 2017. 2

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 6

[22] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001. 2

[23] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011. 2

[24] Z. Wang and J. Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):17, 2015. 9

Are All Training Examples Created Equal? An Empirical Study

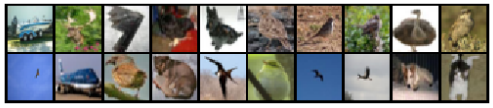
Supplementary Material

Kailas Vodrahalli
UC Berkeley
kailasv@berkeley.edu

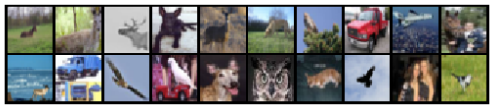
Ke Li
UC Berkeley
ke.li@eecs.berkeley.edu

Jitendra Malik
UC Berkeley
malik@eecs.berkeley.edu

(a) Linear model

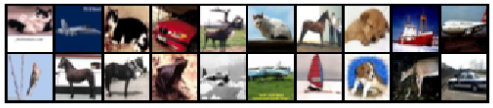


(b) CNN model with vanilla architecture

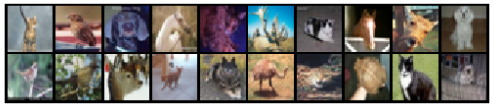


(c) VGG16

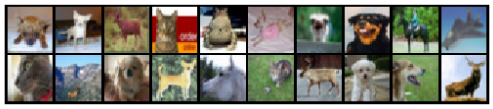
Figure 10: Images from CIFAR-10 with largest gradient magnitudes, organized by network architecture. Note that these images share regularities: for the linear model, the images all have white background; for the vanilla CNN model, images of distant birds appear more commonly than usual. The presence of regularities in terms of pixel values (in the case of the linear model) or semantic categories (in the case of the vanilla CNN model) among images with high gradient magnitudes suggests that selecting images based purely on gradient magnitudes will result in correlated training examples.



(a) Linear model



(b) CNN model with vanilla architecture



(c) VGG16

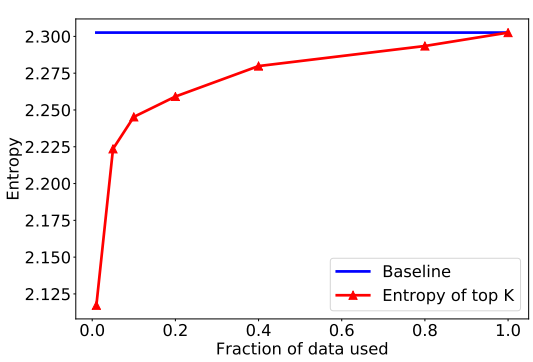
Figure 11: Sample images from CIFAR-10 whose gradient magnitudes are in the top 5%, organized by network architecture. Note the prevalence of cats and dogs in all models, suggesting category-level correlations.

Here we present additional evidence that the examples with largest gradient magnitudes can be correlated. The following results further justify the use of the *Gradient-CDF* method (see Section 3.3).

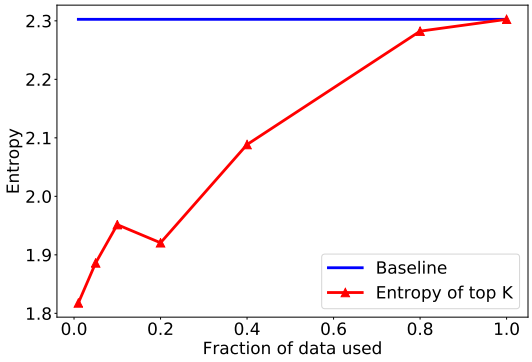
Gradients can be correlated at both the levels of semantic categories and pixels. For category-level correlations, certain categories tend to appear more frequently than others among images with high gradient magnitudes. For pixel-level correlations, certain colours tend to appear more frequently than others among images with high gradient magnitudes.

In Figure 10 we display the images from the CIFAR-10 training set with largest gradient magnitudes in our three models (see Section 4.2). The repeated images of birds in the the vanilla CNN model suggest class-based correlations, and the repeated white backgrounds for the linear model suggest pixel-level correlations.

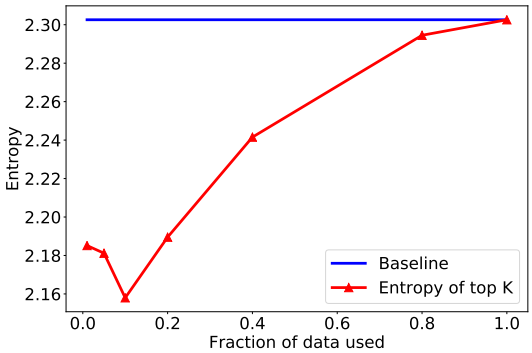
In Figure 11, the prevalence of dog and cat images in both vanilla CNN and VGG16 models suggest class-based correlations. In Figure 12, the low entropy for the CNN model at small subsamples of data indicates that a few classes of images have more images with large gradients than other classes and also suggests class-based correlations.



(a) Linear model



(b) CNN model with vanilla architecture



(c) VGG16

Figure 12: Entropy over class labels for CIFAR-10 images in the top $x\%$ in terms of gradient magnitude for the given trained model. Baseline refers to the entropy for uniformly distributed labels.