Research in Computer Science - Report

Harish Rajagopal
Department of Computer Science
ETH Zürich
20-946-349

Antonio Orvieto
Department of Computer Science
ETH Zürich

Jonas Kohler
Department of Computer Science
ETH Zürich

Xiang Li
Department of Computer Science
ETH Zürich

Prof. Thomas Hofmann
Department of Computer Science
ETH Zürich

Abstract

With the large amounts of data used to train deep neural networks, errors in labelling the data cannot be ignored. However, while label noise is present in numerous datasets, most studies analysing the behaviour of deep neural networks do not consider its impact on their inferences. In this report, we study the effects of label noise on the optimization process of deep neural networks, by training a CIFAR10 classifier with varying levels of artificially added uniformly-distributed label noise. We notice that when trained with learning rate schedulers, the optimal batch size reduces with increasing label noise. We trace this phenomenon to two main observations: uniformly-distributed label noise can lead to non-isotropic gradient noise, and this is often in the direction that leads to the model overfitting on the noisy labels. Further, we also show that learning rate scheduling in the presence of label noise brings about a situation where the commonly-held idea that the ratio of the batch size and learning rate predict generalization is disproven.

1 Introduction

Deep neural networks are well-known to require a large amount of data for training them. This necessitates data collection at large scales, which is often collected through methods that are not 100% accurate, such as manual human labelling. These errors can manifest themselves in the form of 'label noise', which denotes the amount of erroneous ground-truth labels for supervised learning tasks. Since it is highly likely that most datasets contain some amount of label noise, understanding the impact of label noise on neural networks is of great importance.

In this project, we propose to study the impact of label noise on the optimization process for deep neural networks. Optimizing the parameters of deep neural networks is well-known to be a challenging task, and the presence of label noise only complicates this issue. We limit our experiments to showcase how label noise affects some popular beliefs and best practices that researchers have regarding stochastic gradient descent for deep learning, in hopes that this shines new light on some underlying assumptions that researchers have that might not hold, and prompts new research in this direction.

Our findings regarding training deep neural networks in the presence of label noise are:

- **Label noise induces non-isotropic noise in gradients**: Even if the noise is uniformly distributed across all labels, the gradients tend to drift away from the noise-free gradient direction.
- **Training with a larger batch sizes increases overfitting**: This observation shows that the gradient directions are shifting in the direction where the model overfits on the noisy examples, and larger batch sizes accentuate this effect.
- **The optimal batch size can reduce with higher amounts of label noise**: This is a straightforward consequence of the previous two findings, which counteracts the reduction of stochastic gradient noise associated with higher batch sizes.
- **The ratio of batch size to learning rate is not a good indicator of generalization in the presence of label noise and scheduling**: This violates the common belief that higher values of this ratio predict better generalization.

2 Related Work

Rolnick et al. [1] analyse how deep neural networks generalize in the presence of label noise. They show that deep neural networks have a high accuracy, even in the presence of large amounts of label noise. Further, they show that the minimum training dataset size for effective training increases with the amount of label noise. Finally, they claim that higher amounts of label noise reduces the "effective batch size", and that higher batch sizes can mitigate the effects of label noise, due to isotropic gradient noise. However, our experiments show that this last claim does not necessarily hold in practice.

Thulasidasan et al. [2] introduce a loss function that permits the neural network to abstain on incorrectly-labelled samples and learn on the correctly-labelled ones. They further show that when the label noise is correlated with underlying features of the data, training with abstention can lead to the network learning which features are associated with unreliable labels. Although our work does not provide such a definitive solution to learn in the presence of label noise, we show how tuning the right behaviours can mitigate the effects of noisy labels, enabling networks to effectively abstain from noisy examples.

Damian et al. [3] demonstrate that SGD with label noise has an implicit regularization effect. They go on to formulate the regularizer and the regularization parameter, which shows how the regularization effect varies with the batch size, the step size, and the variance of the label noise. In particular, they show that the regularization parameter is inversely proportional to the batch size, which explains our findings about the overfitting effect of larger batch sizes on noisy examples. However, some of our observations cannot be fully explained by their regularization framework.

3 Method

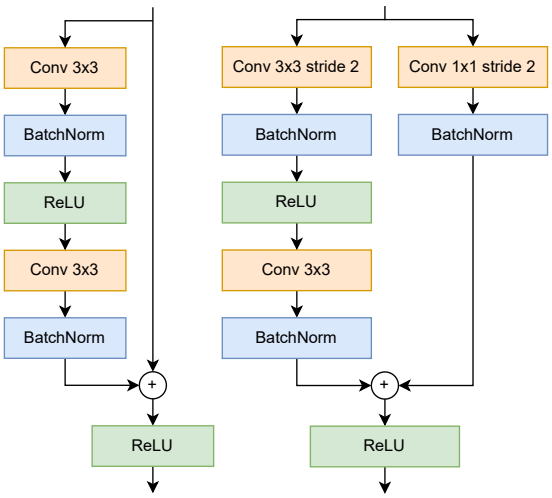
For our analysis, we choose the task of image classification using the CIFAR-10 dataset [4]. The classification model is based on the ResNet-18 [5] architecture. The architecture of the entire model is shown in Figure 2, with architectures of the residual blocks shown in Figure 1.

This model is trained from scratch using the cross-entropy loss along with weight decay using L2 regularization. For optimizing this loss, we use SGD with momentum.

The following three schedulers are used with the optimizer:

- Constant learning rate and momentum, i.e. no scheduler
- Cosine annealing [6], without restarts, for the learning rate only
- One-cycle learning rate [7], for both the learning rate and momentum

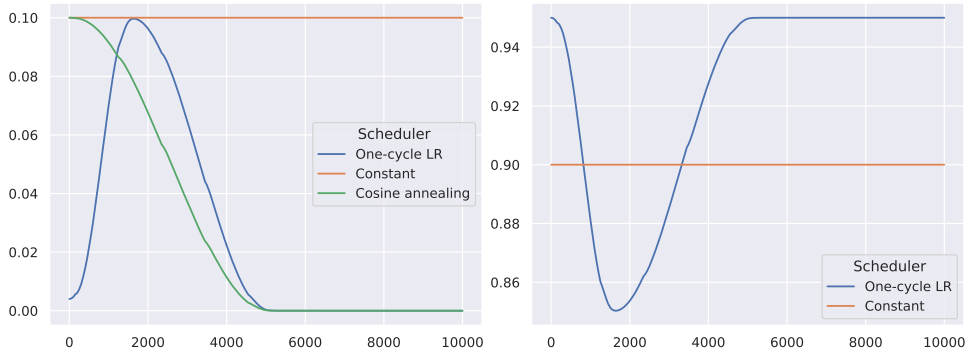
These schedulers are compared in Figure 3. Note that the learning rate and momentum are kept constant after the pre-defined maximum steps for the schedulers is reached.



(a) ResBlock (b) ResBlockDown

Figure 1: Architectures for residual blocks used in the CIFAR10 classifier with code names used in Figure 2

Figure 2: The ResNet18 architecture for the CIFAR10 classifier



(a) Learning rate (b) Momentum

Figure 3: Comparison of schedulers for different hyperparameters

To simulate label noise, we add uniformly distributed noise to the labels. Here, label noise of X% is added by selecting X% of the training dataset. The label for each example in this subset is uniformly sampled from the list of possible classes, including the original class, and permanently changed for the duration of the training procedure.

The common hyperparameters are shown in Table 1. We chose these hyperparameters manually, i.e. without any automatic hyperparameter tuning procedure. Note that the momentum given here is overridden by the one-cycle scheduler. For other hyperparameters, we use PyTorch [8] 1.9.1 defaults.

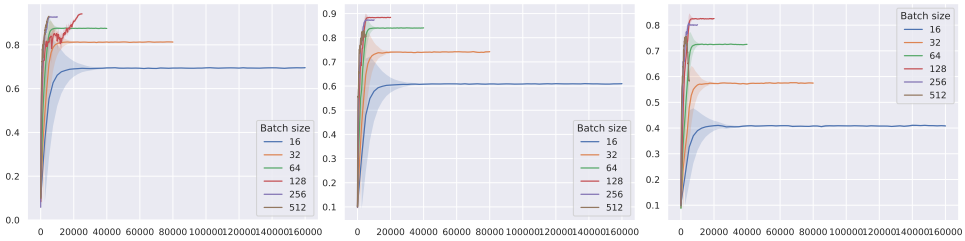
Table 1: Hyperparameters

Name	Value
Maximum learning rate	0.1
Maximum epochs	64
Momentum	0.9
Maximum scheduling steps	5000
Fraction of training dataset for validation	0.2
Weight decay	0.0005

4 Results

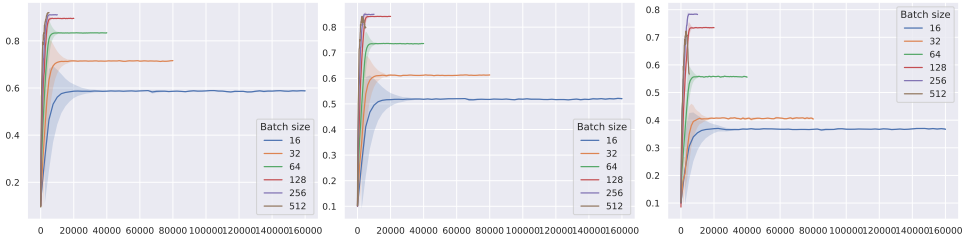
4.1 Generalization with label noise and optimal batch size

Plots for validation accuracy by time for each scheduler are shown in figures 4, 5, 6.



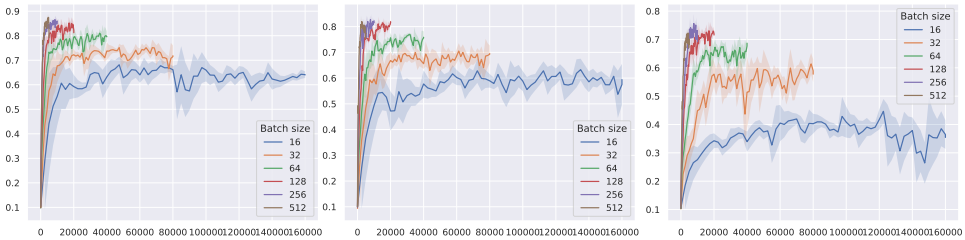
(a) 0% label noise (b) 20% label noise (c) 50% label noise

Figure 4: Validation accuracy with the one-cycle learning rate and momentum scheduler.



(a) 0% label noise (b) 20% label noise (c) 50% label noise

Figure 5: Validation accuracy with the cosine annealing scheduler.



(a) 0% label noise (b) 20% label noise (c) 50% label noise

Figure 6: Validation accuracy with a constant learning rate.

For the two schedulers with 0% label noise, the higher the batch size is, the higher the validation accuracy. However, on increasing the label noise to 20%, the best batch size is no longer the largest

one. Furthermore, increasing the noise to 50%, we see that sometimes the best batch size is not even the second largest.

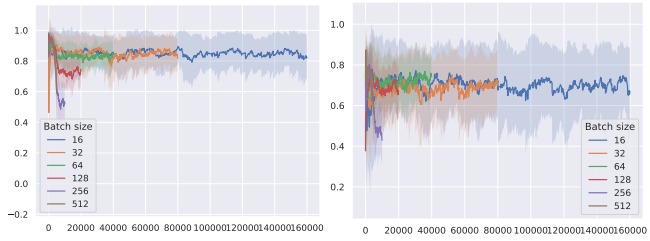
This is in contrast to the claim made by Rolnick et al. [1], who state that increasing label noise decreases the effective batch size. Thus, they claim that increasing batch size mitigates the effect of label noise. However, our experiments show otherwise when a learning rate scheduler is used. Although, for the case of constant learning rates, the claim holds.

4.2 Effect of label noise on gradients

We also plot the cosine similarity between the gradients of clean examples (where the label was not changed) and the overall gradients in figures 7, 8, 9.

(a) 20% label noise (b) 50% label noise

Figure 7: Gradient cosine similarity between all examples and clean examples, with the one-cycle learning rate and momentum scheduler.



(a) 20% label noise (b) 50% label noise

Figure 8: Gradient cosine similarity between all examples and clean examples, with the cosine annealing scheduler.

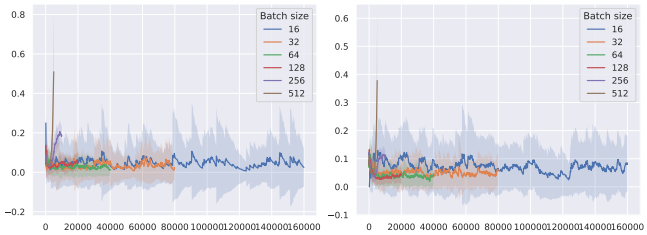
(a) 20% label noise (b) 50% label noise

Figure 9: Gradient cosine similarity between all examples and clean examples, with a constant learning rate.

At a certain time step, the cosine similarity is generally lower with higher batch sizes. Thus, we can infer that the gradients due to noisy examples are non-isotropic when a learning rate scheduler is used, which accumulate with higher batch sizes.

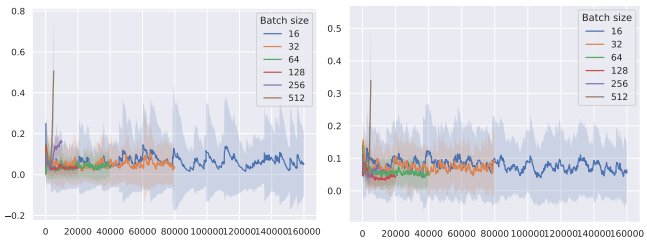
4.3 Overfitting on label noise

We plot the per-batch training accuracy on noisy examples (examples where the label was changed due to label noise) by time for each scheduler in figures 10, 11, 12.



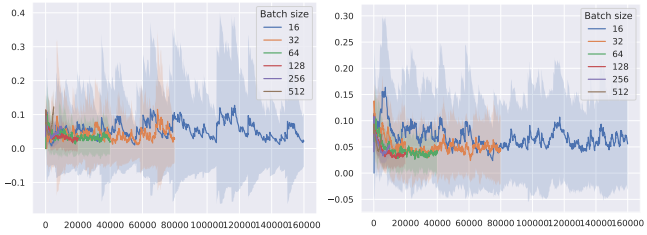
(a) 20% label noise (b) 50% label noise

Figure 10: Training accuracy for noisy examples, with the one-cycle learning rate and momentum scheduler.



(a) 20% label noise (b) 50% label noise

Figure 11: Training accuracy for noisy examples, with the cosine annealing scheduler.



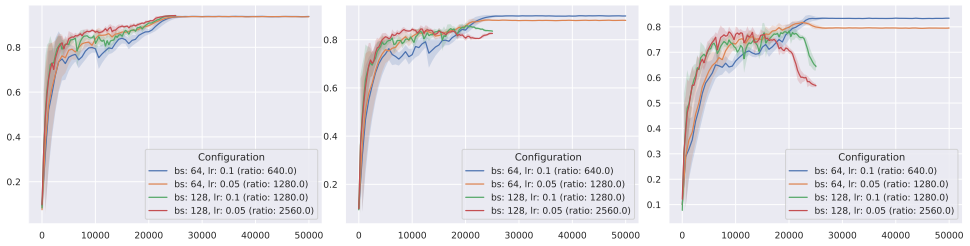
(a) 20% label noise (b) 50% label noise

Figure 12: Training accuracy for noisy examples, with a constant learning rate.

We observe that for a fixed value of label noise, increasing the batch size can lead to overfitting on the noisy examples. This also explains why the generalization suffers with higher batch sizes, as shown in Section 4.1.

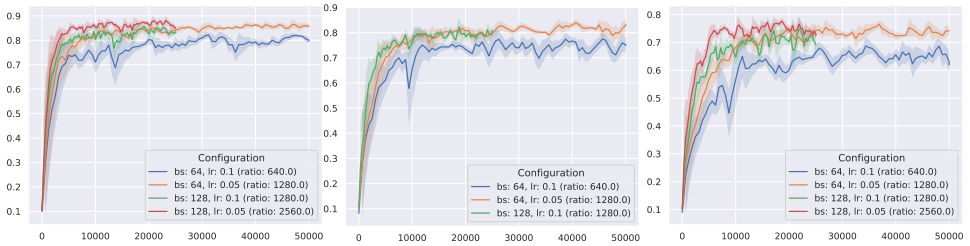
4.4 Ratio of batch size and learning rate with schedulers

Plots for different configurations of batch and learning rate, with the ratio of the batch size to the learning rate, are shown in figures 13, 14.



(a) 0% label noise (b) 20% label noise (c) 50% label noise

Figure 13: Different ratios of the batch size and learning rate with the one-cycle learning rate and momentum scheduler.



(a) 0% label noise (b) 20% label noise (c) 50% label noise

Figure 14: Different ratios of the batch size and learning rate with a constant learning rate.

According to Jastrzębski et al. [9], the ratio of the batch and learning rate should dictate the model's performance, with higher ratios leading to better generalization. We confirm this through the plots for the configurations with either a constant learning rate or zero label noise. However, when using the one-cycle scheduler in the presence of label noise, the configurations with the same batch sizes perform similar, with higher learning rates slightly improving performance. This difference is magnified with higher label noise. Therefore, we find that label noise in the presence of learning rate scheduling can violate this principle. Thus, further analysis in this direction is necessary to refine it.

5 Conclusion

Unlike most studies on the optimization of deep neural networks, we study the generalization of neural networks w.r.t. all the following: label noise, stochastic noise (through batch sizes), and learning rate scheduling. We observe that when a learning rate scheduler is used, increasing the batch size after a threshold can lead to poorer generalization when label noise is present. This behaviour occurs due to non-isotropic gradients induced by noisy examples, which leads to overfitting on these noisy examples, and thus worsens validation accuracy. This contradicts previously-made claims about the behaviour of large batch sizes with label noise. Further, while we empirically verify the theory that the ratio of the batch size to the learning rate predicts generalization, we also come up with a counter example, which is the case when learning rate schedulers are used for a dataset with noisy labels.

While some observations can be explained by some existing literature, further investigation is necessary to theoretically show why this phenomenon occurs. Another limitation of our study is that we have not performed thorough hyperparameter tuning for all configurations, which might have an impact on some of the results. In addition, experiments are also needed to showcase these phenomena for various other classification tasks.

References

[1] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise, 2018.

[2] Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention, 2019.

[3] Alex Damian, Tengyu Ma, and Jason D. Lee. Label noise sgd provably prefers flat global minimizers, 2021.

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[6] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017.

[7] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[9] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd, 2018.

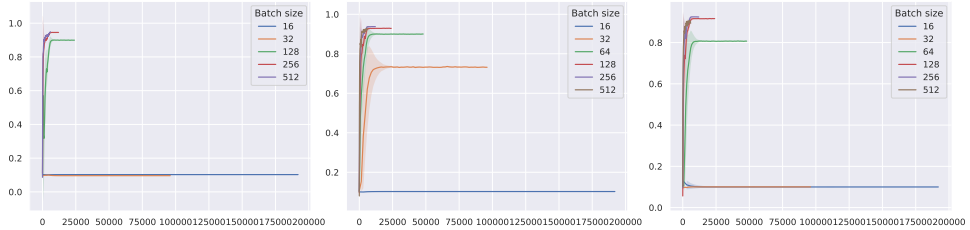
[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[12] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

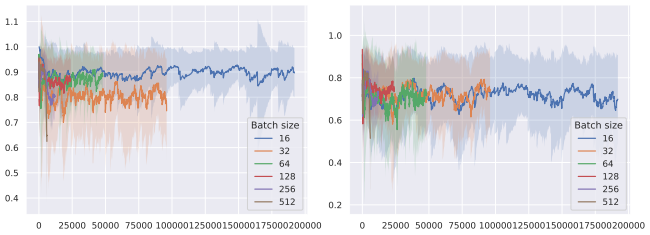
A Image classification with VGG-16 on Fashion MNIST

A VGG-16 [10] model with batch normalization [11] is also trained for the Fashion MNIST [12] dataset. Plots for validation accuracy by time for the one-cycle scheduler are shown in Figure 15. Plots for the cosine similarity between the gradients of clean examples and the overall gradients are shown in Figure 16. Plots for the per-batch training accuracy on noisy examples are shown in Figure 17. However, note that neither were all runs were completed nor were the hyperparameters tuned properly, due to lack of time.



(a) 0% label noise (b) 20% label noise (c) 50% label noise

Figure 15: Validation accuracy on Fashion MNIST using VGG-16.



(a) 20% label noise (b) 50% label noise

Figure 16: Gradient cosine similarity between all examples and clean examples, on Fashion MNIST using VGG-16.

(a) 20% label noise (b) 50% label noise

Figure 17: Training accuracy for noisy examples, on Fashion MNIST using VGG-16.