Semi-supervised Learning and Robustness on a Smaller Scale

Tyler Jiang
Brown University

Abstract

Breakthroughs in semi-supervised learning have enabled large scale models to improve significantly on image classification tasks such as ImageNet-1k. These papers often leverage hundreds of millions or billions of unlabeled images as extra training data to boost performance. In this study, I examine whether the approach of one of these papers, "Self-training with Noisy Student", can boost a model's ability to classify images and robustness to noise when applied at a smaller scale.

1 Introduction

Modern computer vision models achieve exceptional performance on image classification tasks, with the current state-of-the-art achieving a top-1 accuracy of 88.65% on ImageNet-1k. In addition, many of these models have also demonstrated high accuracy on noisy images, such as scaled, rotated, and blurred images. A great example of such a model is Noisy Student, which achieved an impressive 88.4% accuracy on ImageNet at the time of its publication, improving upon the state-of-the-art by 2%. The model also demonstrated remarkably high robustness to adversarial datasets and FGSM attacks without directly optimizing for it.

However, training a model like Noisy Student requires an inordinate amount of resources – Noisy Student required an EfficientNet-L2, 14M images from ImageNet, 300M unlabeled images from an internal Google dataset, and 6 days of training on a Cloud TPU with 2048 cores. Requiring such a substantial amount of resources to achieve their results begs a simple question – which of these improvements can be attributed to breakthroughs in experimental design, and which are simply the results of larger models and more compute? Can the Noisy Student procedure be used in smaller models with a smaller amount of data? Would these models show similar robustness to noise?

Tackling these questions are not trivial – scaling down the task requires staying true to the motivations of the paper. Semi-supervised learning and self-training are often beneficial due to their scale since they rely on an abundance of unlabeled data or the high cost of manual labeling. In addition, previous work has shown that even "large" models like a ResNext-101 32x4d or 32x8d will underfit the training set if it is on the scale of billions of images [Mahajan et al., 2018].

However, work in this area is important because it highlights how feasible new deep learning approaches are for researchers with less access to vast computational resources. For a procedure such as Noisy Student to be useful, it also needs to be accessible – if every researcher needed a cloud TPU and a dataset of 300M+ images for this procedure to work, then it would be infeasible for others to apply it. Additionally, demonstrating that Noisy Student works on a smaller scale directly supports the merits of the procedure itself and opens the door for others to build off of it.

To test these questions, I trained smaller ResNet-50 models on the Caltech-UCSD Birds 200 (CUB-200) dataset and evaluated their performances. These included a fully supervised "teacher" model, a semi-supervised "student" model, and a semi-supervised Noisy Student model. In my results, I show that a smaller model's accuracy benefits significantly from psuedo labeled data and that the Noisy Student procedure boosts robustness to noise, but may not necessarily increase performance overall.

Final report for CSCI 2952-C. Code available at `https://github.com/tyj144/noisystudent-lite`.

2 Background

Since its inception in 2010, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has been the premier benchmark for evaluating image classification performance. ImageNet itself is a database with over 14 million images and over 20,000 categories. As accuracy on this task has increased, the new state-of-the-art models have consistently been set by teams from Facebook AI and Google Research. And this is for good reason – both companies have shifted their focus towards leveraging their access to hundreds of millions if not billions of unlabeled images to improve their models. This is a strategy enabled both by their abilities to construct large datasets and train models at scale, and their growing focus in the fields of semi-supervised and weakly supervised learning.

The motivation for Noisy Student is learning how to leverage large amounts of noisy, labeled and unlabeled data to improve classification performance on ImageNet. The basic procedure is as follows:

1. Train a teacher model on all labeled images in the training set.
2. Use the teacher to generate pseudo labels for unlabeled images.
3. Train an "equal-or-larger" student model on the labeled and pseudo labeled images.
4. Turn the student into the new teacher model and iterate.

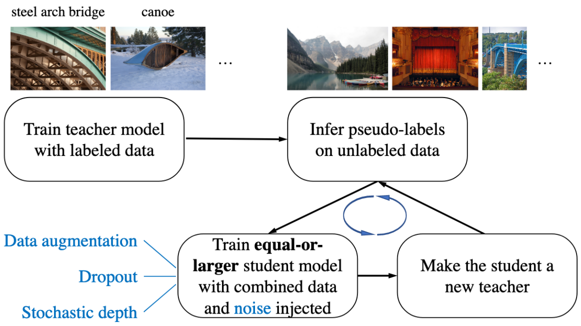
steel arch bridge
canoe
...
...
Train teacher model with labeled data
Infer pseudo-labels on unlabeled data
Data augmentation
Dropout
Stochastic depth
Train **equal-or-larger** student model with combined data and noise injected
Make the student a new teacher

Figure 1: Illustration of the Noisy Student training procedure from the original paper. [Xie et al., 2020]

In Noisy Student, the authors use a standard training procedure, i.e. minimizing the cross-entropy loss on labeled images, to train a teacher model to predict the classes of images from ImageNet. Then, they use the trained teacher model to predict classes of previously unseen, unlabeled images from JFT-300M, Google's internal image dataset of 300 million images. For each image, this prediction is used to form a "pseudo label." There are two kinds of pseudo labels – "soft" or "hard." A "soft" pseudo label is a probability distribution over the classes, and a "hard" pseudo label is a single class – the class of highest probability in the prediction. In Noisy Student, the authors note no significant difference between using soft and hard pseudo labels, and in this report, I use hard pseudo labels.

The student model is then trained on both the labeled images from ImageNet and the pseudo labeled images from JFT-300M. To do this, the authors construct and minimize a custom loss function which sums the average cross entropy losses of the labeled and unlabeled images. This will be discussed further in the "Methods" section.

A key part of this setup is the use of an "equal-or-larger" student model to take advantage of the extra data. Note that both the labeled and unlabeled images are deliberately noised to encourage learning invariance to image transformations such as rotations, translations, shears, and more. The increased difficulty of the student's learning process is noted as a key to improving over the teacher.

The success of Noisy Student in the paper indicates that it is due to the Noisy Student method itself, and not more data or increased model capacity. These results can be summarized as follows:

More data is important, but it isn't everything. It makes intuitive sense that leveraging more unlabeled data will improve model performance. However, work by Mahajan et al. trained a

ResNeXt-101 WSL on 3.5B weakly labeled images, more than 10x the number of images in JFT-300M, but performed worse by 2% and between 10 to 20% worse on various robustness benchmarks. [Mahajan et al., 2018] [Xie et al., 2020] This implies that other factors, like the procedure itself, the difference in architecture, or simply the 2 year difference in compute resources may have been responsible for the improvement.

Bigger models are important, but they aren't everything. Similarly, increased model capacity, when trained on massive (billion-plus) datasets have been shown to boost performance. However, model size itself is not enough. In the Noisy Student paper, the EfficientNet-L2 trained without Noisy Student achieved only 85.5% accuracy, while the EfficientNet-L2 model with it achieved 88.4% accuracy. This 2.9% increase in accuracy implies that adding the Noisy Student procedure boosts model performance in a meaningful way. [Xie et al., 2020]

3 Methods

The methods used in this study are similar to those described in the Background section, with a few key differences. Much of what is included here is derived from Section 2 of the Noisy Student paper. [Xie et al., 2020]

Formally, given labeled images $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and unlabeled images $\{\widetilde{x_1}, \widetilde{x_2}, ..., \widetilde{x_m}\}$ we train a teacher model on the standard cross entropy loss over the labeled examples, i.e.

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i, \theta^t))$$

where $f(x_i, \theta^t)$ is the prediction made on image x_i by the teacher, a model with parameters θ^t.

Then, the teacher generates a pseudo label, $\widetilde{y_i}$, for each unlabeled image, $\widetilde{x_i}$. We then train the student model on a sum of the cross entropy losses on both the labeled and pseudo labeled data. The formula for the student loss function is:

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i, \theta^s)) + \frac{1}{m}\sum_{i=1}^{m} \ell(\widetilde{y_i}, f(\widetilde{x_i}, \theta^s))$$

Here, we average and sum both loss functions to balance the losses of labeled and pseudo labeled data. This gives us a loss function for the student model. When we train the Noisy Student model, we simply noise the input images before making any predictions, i.e. we replace $f(x_i, \theta^s)$, with $f^{noised}(x_i, \theta^s)$, the output of the student on the noised version of x_i. Note only the labeled images are noised in this study, which will be discussed later. To analyze robustness, we apply RandAugment to our test set and evaluate its performance on each model.

The two notable differences between this procedure and the original are how noise is applied and the lack of iterative training. These are goals left for future work to limit the scope of the project. In this study, we will only focus on input noise applied via RandAugment.

4 Experiments

The experimental setup is intended to be as faithful to the original Noisy Student paper as possible. However, to scope out the project and scale down the work, this project has a few key differences described below.

4.1 Experimental Setup

Labeled Dataset. In place of the ImageNet 2012 ILSVRC challenge prediction task, I used the CUB-200 2010 dataset as the labeled set. This is an image dataset of 200 types of bird images commonly used as a fine-grained image classification task.

Unlabeled Dataset. In place of JFT-300M, I used the CUB-200 2011 dataset as the unlabeled set. This is an extended version of the CUB-200 2010 dataset, with double the number of images. I picked this dataset since it is a common image classification task with enough additional examples to

Table 1: Comparison between Noisy Student (Google Research) and Noisy Student Lite (this report).

	Noisy Student	**Noisy Student Lite (this model)**
Model	EfficientNet-L2	ResNet-50
Labeled Dataset	ImageNet	CUB-200 (2010)
Size of Labeled Set	10M images, 10k+ classes	∼6k images, 200 classes
Unlabeled Dataset	JFT-300M	CUB-200 (2011)
Size of Unlabeled Set	300M images, 81M after de-dupe (30x, 8x after de-dupe)	∼12k images (2x)
Noise	RandAugment on labeled and unlabeled, dropout and stochastic depth	RandAugment on the labeled dataset

improve if labeled correctly. Upon review, there does seem to be some overlap between the two sets, but I concluded that the overlap was not significant enough to cause issues or outweigh the benefits, such as the domain fit with the labeled dataset.

In each run, the dataset follows a train-test-validation split of 70-20-10, i.e. 70% of the data is used for training, 20% for testing, etc. The only model which did not was the teacher model that pseudo labeled images for the student, which used a 70-30 train-validation split, but was otherwise not used. I also fixed the data split by fixing the random seed of the data loader, which is used to train and evaluate each model. Each image is read in, converted to a tensor, randomly resized and cropped to a 224 x 224 image, normalized, and randomly flipped in the horizontal direction. Pseudo labeling occurs directly before prediction time, i.e. before a batch is predicted by the student, it is first evaluated by the teacher to generate pseudo labels.

Architecture. I used a ResNet-50 for both the teacher and student models due to its ease of use and availability through the PyTorch "model zoo." These models can load weights pre-trained on ImageNet out of the box, which allowed me to get up and running quickly. To enable the model to classify birds instead of ImageNet classes, I replaced the last layer of the model with a linear layer of size 200 – the number of classes in CUB-200 – and fine-tuned the entire model on the dataset.

Training details. In each reported run, I consistently used the same hyperparameters – 100 epochs, a batch size of 32, an SGD optimizer with a learning rate of 0.01 and momentum of 0.9, and a learning rate scheduler with a decay rate of 0.1 every 30 steps. Additionally, during student and Noisy Student training I split the 100 epochs into 50 epochs for the pseudo labeled set and 50 to fine tune on the labeled set. These were determined after many experimental runs with different hyperparameters.

Noise. I applied RandAugment which randomly applies transformations such as translations, rotations, and shears to the images. In this project, I used an unofficial version of RandAugment which functions the same way as the official version, but had limited customizability. Due to limitations around the pseudo labeling process, I chose only to noise the labeled images. This was because RandAugment could only be applied when reading in an image, i.e. before the teacher could pseudo label it, so it would not provide accurate pseudo labels for the student.

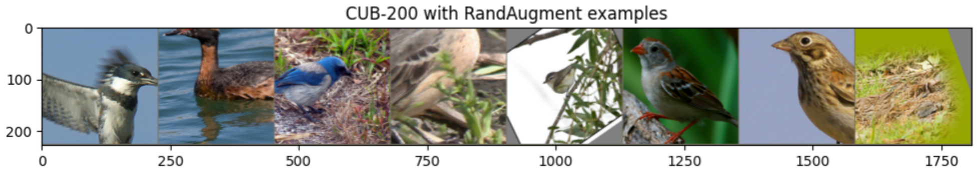

Figure 2: Example of CUB-200 2010 images with RandAugment applied.

4.2 Results

4.2.1 Results on CUB-200

Out of the teacher, student, and Noisy Student models, the highest accuracy was achieved by the student model with a test accuracy of **78.4%** and a maximum validation accuracy of 81% across all epochs. This was a considerable step up from the teacher, which leveled off around 64.0% test accuracy on average. The Noisy Student model, which was fine-tuned on the noisy labeled images, still outperformed the teacher significantly at 77.0%, but fell short of the regular student model.

Table 2: Model accuracy on target task, CUB-200 2010.

Model	Val. accuracy	Test accuracy
Teacher *(ResNet-50)*	68.8%	64.0%
Student *(ResNet-50)*	**81.3%**	**78.4%**
Noisy Student *(ResNet-50)*	N/A	77.0%

The best validation accuracy for Noisy Student is not reported in Table 2, since its validation set was derived from the labeled set with RandAugment, which is used for Table 3. Test accuracies were computed by averaging across five different runs on the same test set, due to noise in the image transformations (e.g. random crop).

The improvement from the teacher to the student and Noisy Student can be attributed to the availability of pseudo labeled data. Since the teacher model was able to generalize over the CUB dataset with 68.8% validation accuracy, it provided an accurate enough set of labels for the student to train on. This provided a total set of labeled and pseudo labeled images triple the size of the original. The model also benefitted from first training on more pseudo labeled examples in the first 50 epochs, then fine tuning on the labeled set. This allowed the model to reach an initial starting point closer to the optimal parameters before training on the labeled set. Noisy Student also benefitted similarly, but it is likely that its performance suffered from the noise in the labeled training data. While it is possible RandAugment scrambled some images too much, e.g. zooming in too far, overall the model likely either did not find a meaningful way to use the noisy inputs to improve or did not see enough instances of regular labeled images.

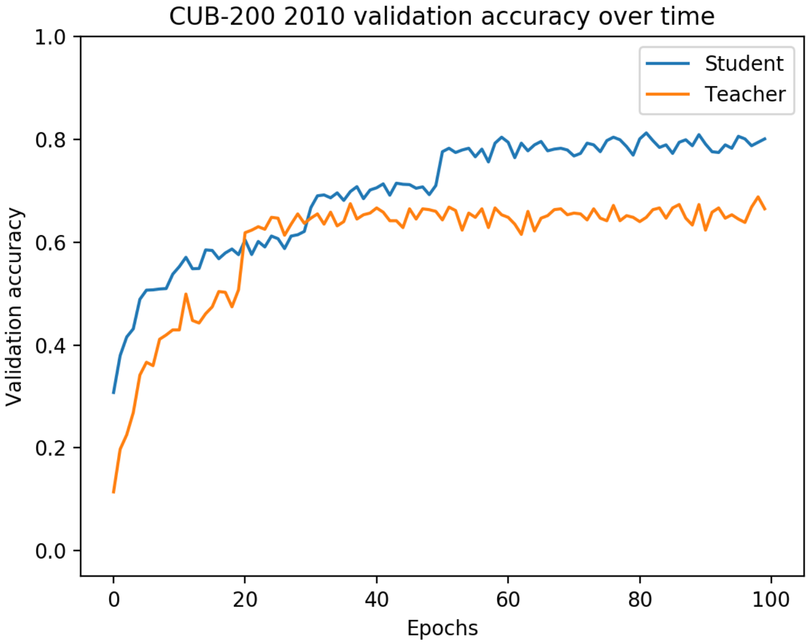

Figure 3: Validation accuracy of teacher and student models on the labeled dataset, CUB-200 2010. Note that the student shifts from pseudo labels to regular labels at epoch 50.

4.2.2 Robustness Results

To evaluate robustness to noise, I applied RandAugment to the labeled data and evaluated the performance of each model on the test set under these transformations. Noisy Student achieved the highest accuracy on the noised dataset with a test accuracy of 66.0%. The method achieved roughly 2.8% better accuracy than the regular student model and a whopping 14.0% better than the teacher model at classifying noisy inputs. This supports the idea that Noisy Student is robust to noise when classifying inputs, although making a definitive conclusion would require evaluating the model on other robustness benchmarks other than the one it was trained on.

Table 3: Model accuracy on robustness task, CUB-200 2010 with RandAugment.

Model	Val. accuracy	Test accuracy
Teacher *(ResNet-50)*	N/A	52.0%
Student *(ResNet-50)*	N/A	63.2%
Noisy Student *(ResNet-50)*	72.1%	**66.0%**

The results of the robustness study suggest there may be a trade-off between model accuracy and robustness to noise, at least specific to the RandAugment transformations used in this project. While the student performed 1.4% better on the target task overall, Noisy Student performed 2.8% on the augmented dataset. This shows that it learned to account for noise when the other models could not, even if it was at the slight expense of overall accuracy. What is more interesting is the regular student model performs substantially better than the teacher on robustness. The regular student model achieved an accuracy of 63.2% compared to the teacher at 52.0%, even though neither model was trained on the noisy inputs. As a result, it is likely that extra data, even when pseudo labeled, can increase robustness. This is likely due to the fact that more examples provide more data and more opportunities to learn, which could help the model detect features in images even when augmented. The increased accuracy could also, more simply, be due to the fact that the student was already better than the teacher at classifying images.

5 Challenges

I faced numerous challenges throughout the process of this project. Initially, I wanted to train the model on a dataset like Tiny ImageNet and evaluate its performance on one of the robustness datasets presented in the original paper, ImageNet-A. However, the models often trained too slowly and maxed out around 30% accuracy for the teacher model, which was not good enough for pseudo labeling.

To remedy this issue, I decided to use a pre-trained model. Since most models are already pre-trained on ImageNet, I shifted my task to another commonly used task, CUB-200. To speed up the training process, I also learned to use GPU training in PyTorch and how to run jobs on the CCV computing cluster. I also spent a lot of time learning the ins and outs of PyTorch and worked to get the teacher model to classify accurately.

Once that was done, I also faced numerous issues with the pseudo labeling process. I initially attempted to construct a custom dataset which mixed labeled and pseudo labeled data into a custom PyTorch Dataset class, but this failed. For a still unknown reason, the teacher model would only predict two or three classes when loaded in via this custom dataset, even though it would predict normally when loaded in other contexts. I also tried to train the model by mixing the data or alternating between labeled and unlabeled, but this often gave poor results. This was later resolved by performing the pseudo labeling right before predicting each batch, but as noted previously, this made it difficult to noise the pseudo labeled images accurately.

6 Related Work

Mahajan et al. performed initial work on weakly supervised learning to pre-train a model on the hashtags of billions of Instagram images, which paved the way for lots of large weakly supervised and semi-supervised procedures. This paper is referred to throughout Noisy Student as the previous state-of-the-art on ImageNet-1k. [Mahajan et al., 2018] The authors are also proponents of larger model sizes to fit their billion-scale datasets. Yalniz et al. also proposed a similar billion-scale semi-supervised learning algorithm which builds the pseudo labeled set by selecting the top K examples from each class. This paper includes a teacher-student procedure very similar to the one in this project. [Yalniz et al., 2019]

Learning robustness to input noise such as blurring, rotating is also a well-studied problem. Zhai et al. showed the effectiveness of self-supervised semi-supervised learning, which attempted to predict how a noisy image was augmented in order to improve image classification accuracy. This approach advocates that learning how to identify the noise in an image enables the model to learn a more useful representation, which could be useful in increasing robustness. [Zhai et al., 2019] Laine et al. proposes using dropout and stochastic augmentation to achieve an ensemble of predictors, similar to the model noise proposed in Noisy Student. [Laine and Aila, 2017] And other studies in language modeling have also shown that larger models do not necessarily improve robustness. [Hendrycks et al., 2020]

7 Limitations

The biggest limitations of this project are:

1. No model noise, like dropout or stochastic depth.
2. No noising of the unlabeled images.
3. No iterative training of the student and teacher models.

During training, the teacher model often ended up at around 89% training accuracy, compared to its validation accuracy of 68.8%. This suggests a fairly high degree of overfitting, which means the model could have benefitted from the dropout or stochastic depth used in the original paper. This would have enabled the model to learn under more difficult circumstances, i.e. zeroing out certain features or dropping layers. Similarly, noising the unlabeled images and pseudo labeling them accurately could have provided more noised examples for the model to learn from, which could have helped it learn a more useful representation. The iterative training could also help, including increasing the size of the student model, but I would first need to address the issue of overfitting.

Other potential limitations of this project include finding a way to customize RandAugment to avoid losing semantic meaning in an image, i.e. by inverting it or cropping/translating too far. It is also worth investigating the overlap in the CUB-200 2010 and 2011 datasets and de-duplicating images to examine how much of an effect this has on the performance.

8 Acknowledgments

I would like to thank Professor Bach for assisting and guiding me throughout the research process, especially in overcoming each roadblock I faced along the way. This research was conducted using the computational resources and services at the Center for Computation and Visualization, Brown University, which I am also immensely grateful for.

9 Conclusion

In this report, I showed that a modified version of the Noisy Student approach works well on the fine-grained image classification task CUB-200. Both the regular student model, trained with the labeled and pseudo labeled data, and the Noisy Student model achieved significant boosts in performance in both overall accuracy and robustness compared to the teacher. The student model outperformed Noisy Student in overall performance with an accuracy of 78.4%, while Noisy Student exhibited consistently higher robustness to noise with an accuracy of 66.0%. Overall, this supports the idea that a semi-supervised approach with additional noise, such as Noisy Student, works on a smaller scale with less data and a smaller model.

References

D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song. Pretrained transformers improve out-of-distribution robustness, 2020.

S. Laine and T. Aila. Temporal ensembling for semi-supervised learning, 2017.

D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining, 2018.

Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification, 2020.

I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan. Billion-scale semi-supervised learning for image classification. *CoRR*, abs/1905.00546, 2019. URL `http://arxiv.org/abs/1905.00546`.

X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer. S4l: Self-supervised semi-supervised learning, 2019.