Little Label Learners Reflection - Mentor Meeting 3

Introduction

Unlike traditional neural networks, infants have limited access to visuals and labels. Self- and semi-supervised learning seek to address this constraint. The former uses no labels, and the latter uses limited amounts to define a latent manifold. We will explore an intermediate: gradual supervision. Starting with pure self-supervision, we will add in more labels over epochs. This will hopefully define richer manifolds than pure self-supervision while using less labels than pure semi-supervision models.

Challenges

A major challenge we have come across in the literature and in our own implementation is determining whether a model is self-supervised, semi-supervised, or fully supervised. Often, the former two are used interchangeably. Other times, fully supervised fine-tuning is given the label of "semi-supervision." Thus, establishing a fundamental definition has been tricky but important. This took us a couple of days, but we are back on track with a good understanding now.

Furthermore, we faced some setbacks working with Oscar. Mostly, this involved setting up a TensorFlow environment with all the needed packages, and critically, PATH variables. When such a setup went wrong, it was not immediately clear; for example, when training our models on a pure CPU environment, we noticed it was faster than its GPU counterpart. This led us to notice the GPU environment did not actually use the GPU. As it turns out, Oscar requires users to use a particular `apptainer`. After a couple of hours of debugging, we have finally solved this issue and can now train new models extremely quickly.

Insights

First, we have managed to preprocess all of our data by creating a `Dataloader` class that handles generating subsets of data containing only a few of the classes of a given dataset. `Dataloader` is also able to split our training data into different proportions of labeled and unlabeled data. This is extremely useful, as our plan to demonstrate gradual supervision requires variable labeling rates, while our goal to show continual learning needs datasets containing different classes.

Furthermore, we have also gathered many baseline results, such as a simple CNN, a self-supervised model, and a semi-supervised model. For these models, we tested their ability to learn novel classes on the same dataset, CIFAR10. Currently, our model is performing mediocrely. Nevertheless, we must still define an appropriate degree of "novel learning", as well as optimal split rate and other hyperparameters.

Lastly, we have worked on analyzing the quality of our latent spaces. This was done through several dimensionality-reduction and visualization experiments, namely involving PCA and UMAP. Currently, they are not as good as we would have wanted, but this is likely due to our overly simplistic model and low epoch-count. However, we are happy that we have these things working and can actually use it to see how our model is working. We can now just focus on changing some things about how we are training our model with our proposed strategy in order to see if we can generate better results than what we currently have.

Plan

At this point in the project, we are confident that we are on the right track to finish on time. However, there are several opportunities to finetune. For example, we would like to explore more complicated CNN encoder architectures, such as VGG16, in order to provide more accurate use cases of our training regime. On the classification side, we are currently using a Sparse CCE Loss to "shortcut" our way to evaluating a limited-class prediction output against a full-class ground truth. Changing our models to handle one-hot encoding and use explicit CCE Loss would better highlight our intentions for our model to learn new classes, and also introduce the opportunity to play with "unknown" class labels.

As for generating our latent space, we are currently using the linear-probe backpropagation to update our encoder's weights. This represents a rather abstract understanding of latent-space "accuracy." Thus, another stretch goal of our's is to develop more interpretable and possibly realistic semi-supervised losses. These will most likely be inspired by existing semi-supervision methods based in nearest-neighbor and pseudo-labeling techniques. Additionally, we would like to make many more measurements on our encoder-generated latent spaces, such as cluster-quality metrics such as ARI and NMI.

Findings Table:

Model	Learning Type	Dataset	Train/Test Data	Max Validation (%)
Baseline	Fully Supervised	CIFAR10	Full train, full test	59.18

Baseline	Fully Supervised	CIFAR10	train 7 classes; full test	43.11
Baseline	Fully Supervised	CIFAR10	train 7 classes; test 7 classes	59.59
Baseline	Fully Supervised	CIFAR10	Train full; test 7	54.97
SimCLR	Self Supervised	CIFAR10	Full train, full test	31.27
SimCLR	Self Supervised	CIFAR10	train 7 classes; full test	25.30
SimCLR	Self Supervised	CIFAR10	train 7 classes; test 7 classes	37.61
SimCLR	Self Supervised	CIFAR10	Train full; test 7	32
Novel Model	Gradual Supervision	CIFAR10	Train begin 7 classes and gradually become 10; full test; Split rate beginning 0.01 and gradually becomes 0.5; split rate increase by 0.05	39.14
Novel Model	Gradual Supervision	CIFAR10	Train begin 7 classes and gradually become 10; full test; Split rate beginning 0.01 and gradually becomes 0.7; split rate increase by 0.05	40.63
Novel Model	Gradual Supervision	CIFAR10	Train begin 5 classes and gradually become 10; full test; Split rate beginning 0.01 and gradually becomes 0.7; split rate increase by 0.05	40.35
Novel Model	Gradual Supervision	CIFAR10	Train begin 7 classes and gradually become 10; test split data; Split rate beginning 0.01 and gradually becomes 0.5; split rate increase by 0.05	40.03
Novel Model	Gradual Supervision	CIFAR10	Train begin 7 classes and gradually become 10; test split data; Split rate beginning 0.01 and gradually becomes 0.7; split rate increase by 0.05	41.43

Novel Model	Gradual Supervision	CIFAR10	Train begin 5 classes and gradually become 10; test split data; Split rate beginning 0.01 and gradually becomes 0.5; split rate increase by 0.05	40.24
Novel Model	Gradual Supervision	CIFAR10	Train full; test full; Split rate beginning 0.01 and gradually becomes 0.7; split rate increase by 0.05	42.25
Novel Model	Gradual Supervision	CIFAR10	Train full; test full; Split rate beginning 0.01 and gradually becomes 0.5; split rate increase by 0.05	41.77
Novel Model	Gradual Supervision	CIFAR10	Train full; test full; Split rate beginning 0.01 and gradually becomes 0.9; split rate increase by 0.05	42.17
Novel Model	Gradual Supervision	CIFAR10	Train full; test full; Split rate beginning 0.5 and gradually becomes 0.7; split rate increase by 0.05	42.92
Novel Model	Gradual Supervision	CIFAR10	Train 7; test full; Split rate beginning 0.01 and gradually becomes 0.5; split rate increase by 0.05	30.71
Novel Model	Gradual Supervision	CIFAR10	Train 7; test full; Split rate beginning 0.01 and gradually becomes 0.7; split rate increase by 0.05	30.78
Novel Model	Gradual Supervision	CIFAR10	Train beginning 9 and the 10; test full; Split rate beginning 0.01 and gradually becomes 0.5; split rate increase by 0.05	41.05
Novel Model	Gradual Supervision	CIFAR10	Train beginning 1 gradually 10; test full; Split rate beginning 0.01 and gradually becomes 0.99; split rate increase by 0.05	38.45

Novel Model	Gradual Supervision	CIFAR10	Train beginning 5 gradually 10; test full; Split rate beginning 0.01 and gradually becomes 0.99; split rate increase by 0.05	40.36

Made some changes to Gradual Supervised Model After Mentor Meeting 3:

Novel Model	Gradual Supervision	CIFAR10	Start training with 10, finish training with 10	50.79%
Novel Model	Gradual Supervision	CIFAR10	Start training with 9 gradually train up to 10, finish training with 10	49.49%
Novel Model	Gradual Supervision	CIFAR10	Start training with 8 gradually train up to 10, finish training with 10	48.46%
Novel Model	Gradual Supervision	CIFAR10	Start training with 7 gradually train up to 10, finish training with 10	47.66%
Novel Model	Gradual Supervision	CIFAR10	Start training with 6 gradually train up to 10, finish training with 10	48.66%
Novel Model	Gradual Supervision	CIFAR10	Start training with 5 gradually	49.27%
Novel Model	Gradual Supervision	CIFAR10	Start training with 4 gradually train up to 10, finish training with 10	47.47%
Novel Model	Gradual Supervision	CIFAR10	Start training with 3 gradually train	48.64%

			up to 10, finish training with 10	
Novel Model	Gradual Supervision	CIFAR10	Start training with 2 gradually train up to 10, finish training with 10	47.14%