Robustness of Contrastive Learning Models

Ryan Campbell, Dragos Cristian Manta, Alexander Iannantuono
Supervised by: Prof. Adam Oberman

September 1, 2020

One of our research topics this Summer is about a new approach in self-supervised deep learning, called *contrastive learning*. All of the applied work was done in the framework of image recognition.

One of the issues of traditional deep learning is the necessity of a large labeled training set, which can often be expensive to get, in order to achieve high accuracy on new samples. Another one is *robustness*, the ability of the model to maintain a high accuracy when receiving images that have been slightly perturbed. Contrastive learning tries to fix both of these problems by proposing a two stage learning process: an unsupervised part followed by a supervised one, as will be outlined below:

Stage 1: Encoder Training

Given a training set of *unlabeled* images $\{x_k\}_{k=1}^T$, we train an encoder neural network to learn an embedding $\Phi : \mathbb{R}^d \to \mathbb{R}^m$, where d and m are the number of dimensions of the original features and of the latent space respectively. The constraint enforced by the new loss, called *normalized temperature-scaled cross entropy loss*, during training, is designed such that the learned map satisfies a key desirable property: images of the same class should be mapped "close" and those of different classes should be mapped "far away". We will soon define what we mean by that. A burning question is: how is this done if we have no access to the class labels at this stage? This is where *data augmentation* comes into play.

Let $\mathcal{T}$ denote the set of data augmentations given by the composition of random crop and resize, random horizontal flip, random grayscale and random color jitter. Given a batch of N images $\{x_k\}_{k=1}^N$, for every $1 \leq k \leq N$, the idea is to draw two augmentations $t, t' \sim \mathcal{T}$ at random and compute $\tilde{x}_{2k-1} = t(x_k)$ and $\tilde{x}_{2k} = t'(x_k)$. This yields $2N$ images. Letting $z_i := \Phi(\tilde{x}_i)$, we define

$$\ell(i,j) := -\ln \frac{e^{\frac{s_{i,j}}{\tau}}}{\sum_{k=1}^{2N} e^{\frac{s_{i,k}}{\tau}}},$$

where τ is a temperature hyperparameter and where $s_{i,j} := \frac{\langle z_i, z_j \rangle}{\|z_i\| \|z_j\|}$ is the similarity metric mentioned earlier. Then, if we define the loss as

$$\mathcal{L} := \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)], \tag{1}$$

minimizing it will maximize the similarity of outputs of Φ for pairs of images that are two augmentations of the same original image and will minimize the similarity in the other cases.

Stage 2: Linear Classifier Training

Since two augmentations of the same image come necessarily from the same class, the encoder trained with this loss should, at the end, map (unseen) images from the same class close together and others far away. Hence, by treating these outputs as new features (those are called *learned representations*), we can train a supervised linear classifier (logistic regression) using these features as inputs. The hope is that fewer labels are needed to achieve the same accuracy since the geometry of the learned representations should help.

Experiments

The applied part of this research project was driven by carrying out many experiments. Some of them were designed to try to confirm the results found in the litterature, while others to test new ideas. Most of them were carried out on the CIFAR-10 dataset, using PyTorch, on 4 GeForce GTX 1080 Ti GPU's. The encoder network used was a modified ResNet50, following the procedure in [1].

To test learning with fewer labels, we trained a supervised ResNet50 model (which we defined as our baseline) for 200 epochs by giving it access to only a certain fraction of the labels from the training set. We then trained the ResNet50 encoder with the contrastive loss on the full training set for 100 epochs (no labels are needed at this stage), then we froze the encoder's weights and trained a logistic regression classifier as described in stage 2 with the same number of epochs, but with access to the same fraction of the labels as the first model. A strong first conclusion that we found is that indeed, contrastive learning models' performances are much less negatively affected by the loss of labels from the training set than their supervised counterpart, as outlined in Table 1.

Terminology. For the upcoming part of the discussion, we refer to *Tikhonov regularization* simply by calling it *regularization*.

To test robustness, we decided to compare the baseline and contrastive models trained on the full set of labels. We also wanted to see how adding regularization to the baseline compares with using the contrastive model without it. To be more precise, we minimized the following loss [see 2]:

$$\mathcal{L}_{tik} = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\mathcal{L}(f(x,w),y) + \frac{\lambda}{2}||\nabla_x \mathcal{L}(f(x,w),y)||^2\right], \tag{2}$$

where we always measured the norms in ℓ^2. In our experiments, we used $\lambda = 0.1$, $\lambda = 1$ (following [2]) and $\lambda = 10$. In order to apply regularization to the contrastive model however, we cannot blindly apply (2) because the contrastive loss (1) depends on all the samples in the batch, as opposed to one loss per sample image in the supervised case. Hence, taking an expectation of a gradient norm is more difficult to justify in the contrastive case, especially since (1) already has the form of an average, thus averaging twice might lead to the penalty term being "dampened". We can experimentally confirm that this is problematic. The new regularized loss that we used is then

$$\mathcal{L}_{tik} := \mathcal{L} + \frac{\lambda}{2}\mathcal{R} := \mathcal{L} + \frac{\lambda}{2}\sum_{k=1}^{T}\frac{||\nabla_{\tilde{x}_{2k-1}}\mathcal{L}||^2 + ||\nabla_{\tilde{x}_{2k}}\mathcal{L}||^2}{2}, \tag{3}$$

where the rationale behind the terms in the sum is to take the average between the squared norms of the input gradients with respect to both augmentations $\tilde{x}_{2k-1}$ and $\tilde{x}_{2k}$ of the same image x_k.

We found very convincing evidence that contrastive models are more resistent to PGD and PLB attacks [see 4], as seen in Figure 2.

A major difficulty that we had, though, is to reproduce some of the high claimed accuracies from the papers (regarding the contrastive model). It must be admitted that training Machine Learning models is often a tricky task riddled with technical difficulties. In addition, papers themselves also don't seem to always follow good practices for experiment reproducibility, as explained in [3].

We also fine-tuned the contrastive model by repeating almost the same procedure as the one described in stage 2, except that we let the weights of the encoder to change as well. Another difference is that, instead of a logistic regression classifier, we used a 2-layered fully connected network with relu activation. This increased the accuracy slightly. We found that randomly initializing the weights yielded the best results. See Table 1 for details about the accuracies.

% of Labels Available at Train Time	Supervised Model	Contrastive Model	Fine-Tuned
100%	93.93%	74.79%	77.05%
10%	64.39%	70.83%	71.36%
1%	30.92%	67.56%	66.94%

Table 1: Test set accuracy on CIFAR-10 of a regular supervised model compared with our contrastive one, both of which were trained with access to only a certain fraction of the labels from the training set. The last column displays the accuracy of the fine-tuned contrastive model.

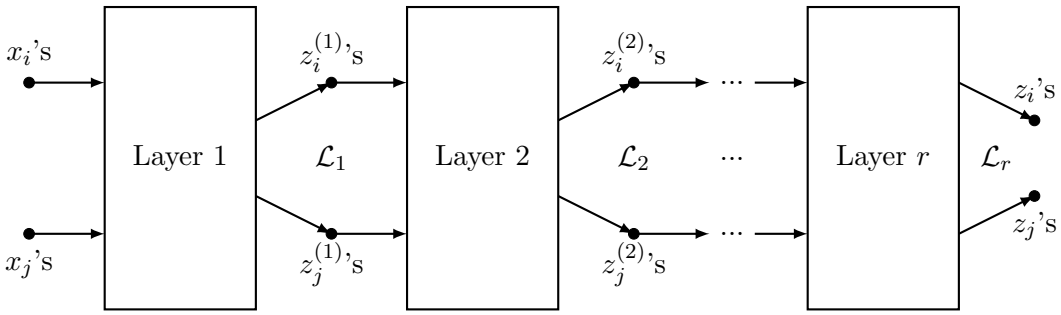


Figure 1: Illustration of the layer-by-layer loss.

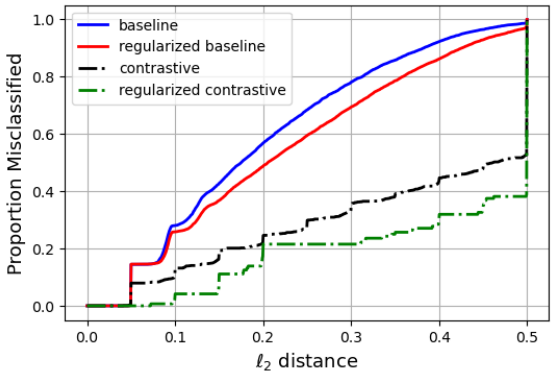
Future Experiments

Some new ideas yet to be implemented involve applying the contrastive loss from (1) to each layer of the encoder. To improve robustness, we could apply Tikhonov regularization layer-by-layer to not only penalize net sensitivity to inputs, but also too big layer-by-layer variations (see Figure 1). The loss would then become

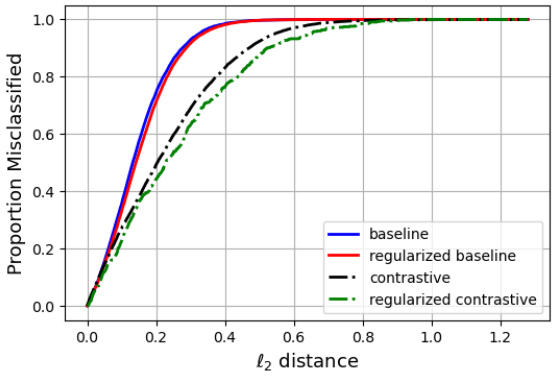
$$\mathcal{L} := \sum_{k=1}^{r} \alpha_k \mathcal{L}_k, \text{ where } \alpha_i \ll \alpha_j \iff i < j \tag{4}$$

and the regularized version would be

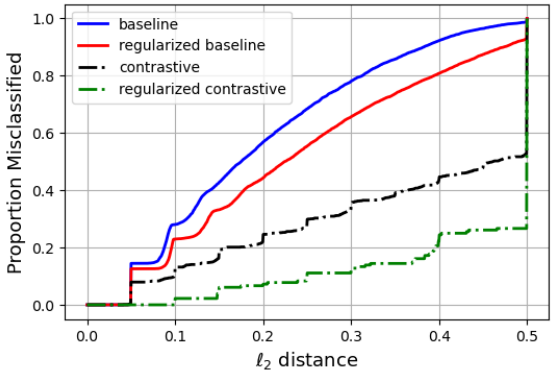
$$\mathcal{L}_{\text{tik}} := \mathcal{L} + \sum_{k=1}^{r} \beta_k \mathcal{R}_k, \text{ where } \beta_i \ll \beta_j \iff i < j \text{ and } \mathcal{R}_i = \sum_{k=1}^{T} \frac{||\nabla_{\tilde{x}_{2k-1}} \mathcal{L}_i||^2 + ||\nabla_{\tilde{x}_{2k}} \mathcal{L}_i||^2}{2}. \tag{5}$$



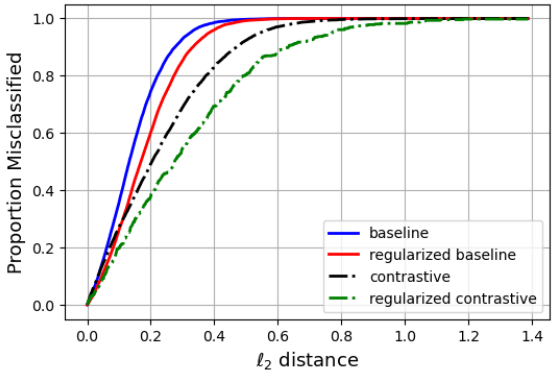
(a) PGD attack, $\lambda = 0.1$



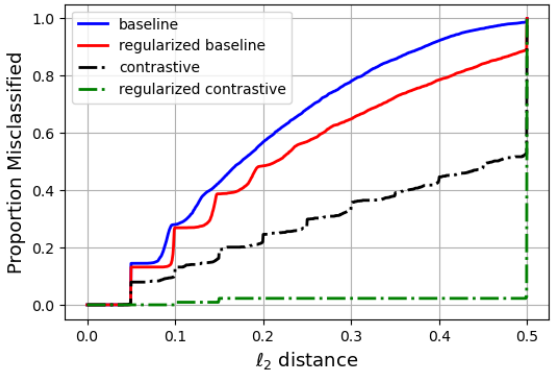
(b) PLB attack, $\lambda = 0.1$



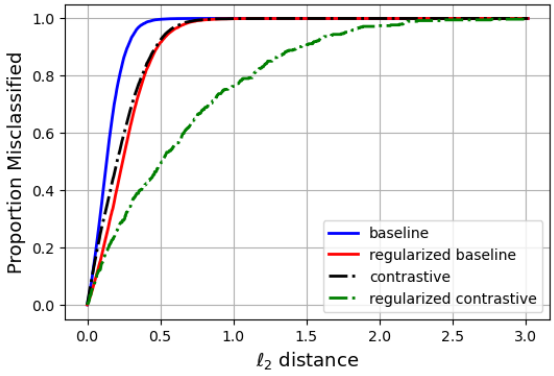
(c) PGD attack, $\lambda = 1$



(d) PLB attack, $\lambda = 1$



(e) PGD attack, $\lambda = 10$



(f) PLB attack, $\lambda = 10$

Figure 2: Percentage of misclassified images after being perturbed. We clearly see that using the contrastive model brings a bigger gain in robustness than applying regularization alone. Furthermore, applying regularization to the contrastive model improves resistance to attacks even more. Hence, combining the two methods dramatically increases robustness. Compared to the original contrastive model, the $\lambda = 0.1$ regularized one only lost 2.89% of accuracy, the $\lambda = 1$ one lost 7.03% and the $\lambda = 10$ one lost 14.08%.

References

[1] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *arXiv preprint arXiv:2002.05709* (2020).

[2] Chris Finlay and Adam M Oberman. "Scaleable input gradient regularization for adversarial robustness". In: *arXiv preprint arXiv:1905.11468* (2019).

[3] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. "A metric learning reality check". In: *arXiv preprint arXiv:2003.08505* (2020).

[4] Aram-Alexandre Pooladian et al. "A principled approach for generating adversarial images under non-smooth dissimilarity metrics". In: *International Conference on Artificial Intelligence and Statistics*. 2020, pp. 1442–1452.