An Empirical Study of Supervised Contrastive Loss for Enhancing CNN Robustness

Lorenzo Cusin, Giacomo Serafini
Github Repo

05/02/2026

1 Introduction

In recent years, Contrastive Loss (CL) [1] has gained relevance in the field of Representation Learning. In fact, it gives models the possibility to encode a specific input in a well-structured, low-dimensional latent space. Several studies have shown its role in improving performance across various tasks, in both supervised [2] and unsupervised [3] learning.

In this project, we want to explore its ability to improve the local robusteness of Convolutional Neural Networks (CNN) in the field of image classification. Usually, the networks' robustness deficit is related to their incapacity to learn relevant features from the input and create a solid decision boundary. By its nature, CL can bring improvement in both cases.

In our setting, we fix the model architecture and we train it on CIFAR10 dataset with different types of strategies: data augmentation, adversarial training with PGD, supervised contrastive training and certified training with CROWN-IBP. To evaluate the robusteness we employed the PGD Attack Success Rate (ASR) and the α, β-CROWN method.

2 Background

In this section, we explain the training methods used and their relative impact on model robustness.

2.1 Data Augmentation

Data Augmentation is the process in which input data are transformed during the training phase. To make an example, common augmentation in the image field are rotations, cropping and light balancing. The purpose of this process is to create slightly different images in order to improve the model's generalization ability.

Robustness Impact. Since we are showing perturbated samples, the model is likely to acquire a better understanding of important features, becoming less sensible to slight variations.

2.2 Adversarial Training

The Adversarial Training is a type of training completely oriented to increase the robustness of the model. During the training process, methods to find adversarial examples, like PGD, are executed on the input data. The obtained malicious input is appended to the training set and the model is trained. It can be considered as a specific type of augmentation.

Robustness Impact. Since the model deals with adversarial examples in its training phase, it naturally increases its robustness against that particular attack type.

2.3 Certified Training

In the field of Neural Network Verification, bound propagation is a tool used to verify the local robustness of a neural network with respect to particular properties. It involves the creation of a perturbated input by an ε value with respect to a specific L_P norm and the propagation of its lower and upper bound through the model. The property is considered verified if, for all inputs within the ε-ball, the propagated bounds guarantee that the properties hold. Typically, this means that the lower bound of the desired output remains above (or the upper bound remains below) the violation threshold; otherwise, it is considered not verified. Methods like CROWN-IBP perform efficiently this process.

Applying Certified Training means perturbing the input during the training phase and considering the lower/upper bound to compute the loss. In the field of image classification, this usually means evaluating the Cross Entropy loss using the relative image lower bound.

In general, this method is able to guarantee a better local robustness paying the price of accuracy degradation.

Robustness Impact. Since the network learns to predict the worst case scenario around several input points, it can increase its robustness to adversarial examples.

2.4 Supervised Contrastive Loss

Given a set of points and their corresponding labels, the Supervised Contrastive Loss (SCL) [2] encourages low intracluster distances and high intercluster distances. As a result, points from the same class are mapped close to each other and far from those of other classes. Its mathematical formulation is stated as:

$$\mathcal{L}_{\text{sup}} = \sum_{i=1}^{N} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\frac{z_i \cdot z_p}{\tau}\right)}{\sum_{a \in A(i)} \exp\left(\frac{z_i \cdot z_a}{\tau}\right)} \quad (1)$$

where:

- N is the total number of samples in the batch;
- z_i is the normalized point i;
- $P(i)$ is the set of positive points for anchor i (all samples in the batch that share the same label as i, excluding i itself);
- $A(i)$ is the set of all points in the batch except i (positives + negatives);
- τ is the temperature parameter controlling the concentration of the distribution.

Let's consider this loss in the field of images classifcation using CNNs, where the points mentioned above can be considered as image embedding vectors. They can be obtained from the model's *encoder*, that is, a sequence of convolutional layers followed by a ReLU based Neural Network. The training process can be performed in two different ways:

1. **Double Training Method (DTM)**: first train the *encoder* using the SCL in order to extract meaningful features from the image and map them in the latent space, then train a *classification head* (e.g. Linear Classifier) with Cross Entropy (CE) loss, taking the embeddings as input;

2. **Single Training Method (STM)**: train the entire network (e.g. encoder + classification head) using a linear combination of SCL and CE:

$$L = \alpha L_{SCL} + (1 - \alpha) L_{CE}$$

In this case α is a hyperparameter to tune.

To better understand the property that SCL is imposing into the latent space, the UMAP representations of the embeddings produced by a CNN's encoder trained with and without contrastive loss are shown:

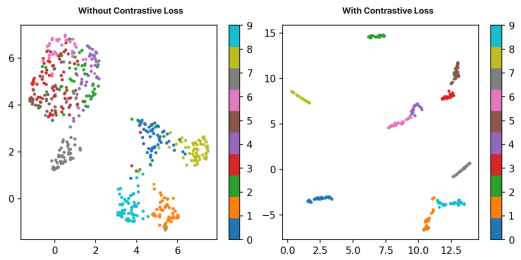

Figure 1: *UMAP representation of 100 embeddings of a model trained with contrastive loss and without it.*

Clearly this is a good scenario that is not always possible to obtain: we show it just to have a graphical understanding of the description made before.

Robustness Impact. Since Adversarial Attacks leverage on weak features learning and loose decision boundaries, it seems logical to say that SPL can bring benefits in the field of network robustness.

3 Experiments

In this section, we are going to describe our settings and considerations of the experiments.

3.1 Hardware Specs

We executed the tests using a Intel(R) Xeon(R) CPU @ 2.20GHz with 30GB of RAM and a GPU NVIDIA T4 Tensor with 15GB of VRAM.

3.2 Dataset

CIFAR10 is the selected dataset. It contains 60.000 $3 \times 32 \times 32$ images related to 10 classes, including animals and vehicles. The train and test set are composed respectively of 50.000 and 10.000 items. It is a good trade-off between complexity and requested computational resources, which is perfect for our purpose.

A validation set is created by splitting the train set with the 80-20 rule, and it is going to be used for hyperparameter tuning.

3.3 Model

The following base architecture is employed:

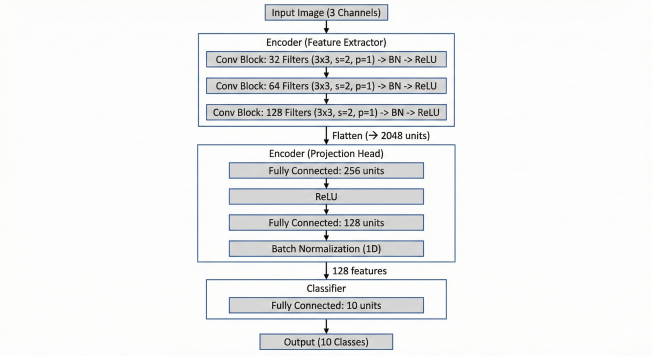

Figure 2: *Model Architecture*

It is not oriented to gain the maximum accuracy: again, we want a good-enough result with a non-expansive computational demand. In the encoder, striding is employed instead of pooling layers to perform dimensionality reduction. This choice is due to the high computational cost requested by α, β-CROWN to manage the pooling layers. Finally, the output of the *encoder* block is considered as the image's embedding.

4 Training

The model architecture is fixed and the following training methods are applied:

1. **Normal Training**: the model is trained using only the Cross Entropy loss;
2. **Adversarial Training**: the model is trained with Normal Training but the images' batch is augmented with Adversarial Examples found by the PGD method ($\varepsilon = 8/255$);
3. **Certified Training**: the model is trained using Normal Training but the loss is evaluated using the lower bounds obtained by CROWN-IBP ($\varepsilon = 2/255$).

For each of them, a version with the augmentation is created. The 6 models obtained are our baselines, so, to obtain the respective *contrastive models*, Supervised Contrastive Loss with Double Training is then applied. SPL with Single Training is avoided for training instability due to the combination of two different losses. The performance of each model is maximized using basic hyper-parameter tuning to learning rate and batch size.

5 Results

In this section, we will refer to each model with the relative training name. For instance, *adversarial model* is the model trained with *adversarial training*; *adversarial contrastive model* is the model trained using the combination of *adversarial training* and the *supervised contrastive training*. Regarding augmentation, the reference will be clarified by the context.

In the following paragraphs, only the result on the test set are shown. Informations strictly related to the training process can be found in the Github repository.

5.1 Accuracy

Here is a summary of the obtained accuracy:

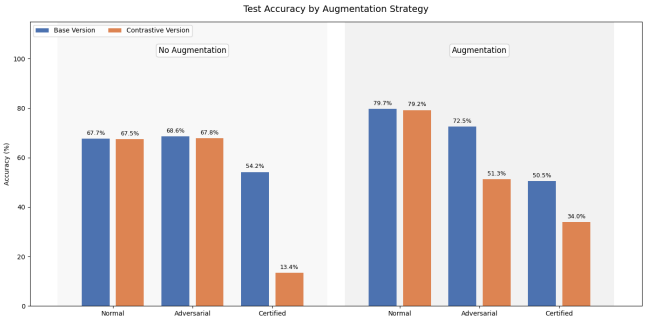

Figure 3: Accuracy of the models

Relevant observations are:

- the augmentation is increasing the performances only for the normal and normal contrastive model. This is expected since they can learn more meaningful feature representations;
- with augmentation, SCL is degradating the accuracy of the adversarial contrastive model. In this case, the batches are composed by the augmented images and their relative adversarial examples. This means that the initial input distribution is highly enlarged and the encoder struggles to map embeddings relative to different class images separatly. As a consequence, the linear classifier is not performing well in the discrimination task. This phenomenon is not present in the adversarial model: encoder and training are working together to maximize the classification ability;
- the certified model, both in the augmentation and no augmentation setting, suffers from accuracy decrease. This is a widely known situation in the literature, so it is expected;
- the certified contrastive model leverages on data augmentation to gain performance but, looking closely, this should not happen because the situation is similar to the one described in point 2. In fact, the initial input distribution should be highly enlarged by the use of the augmented images' lower bound. In this case, the main difference is due to the

ε value: PGD search is performed with $\varepsilon = 8/255$, while CROWN-IBP with $\varepsilon = 2/255$. This reduction is not highly enlarging the initial input distribution allowing the certified model to improve its accuracy.

5.2 Robusteness Analysis

To analyze the robustness, the PGD attack is executed over the entire test set through a variable $\varepsilon \in \{1/255, 2/255, 4/255, 8/255, 16/255\}$. Then, α, β-CROWN is run over 20 images per model in which the previous attack failed.

5.2.1 Robust Accuracy

The success of an adversarial attack is the ability to switch a correct prediction into an incorrect one. So, the Attack Success Rate (or equivalently, the Attack Failure Rate) is evaluated only over all the correct predicted images. To have a clear representation of the performance of the model, the accuracy of each model is weighted by the respective PGD Attack Failure Rate (AFR). We call this metric Robust Accuracy and it is formally defined as:

$$RA = Accuracy \times AFR$$

The results can be summarized by the following plots:

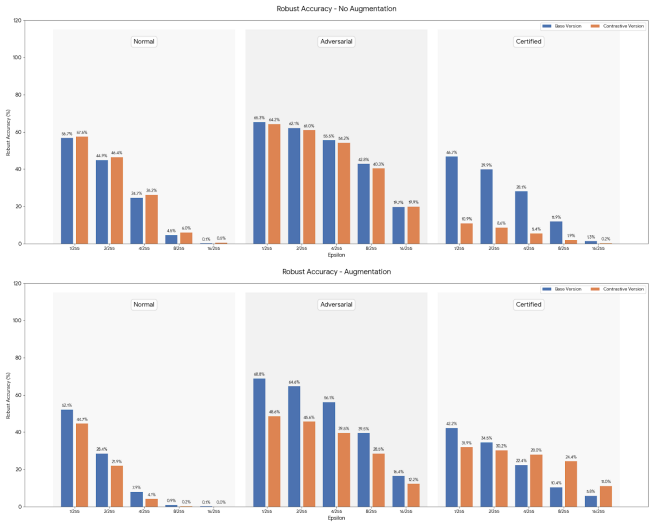

Figure 4: Robust Accuracy for each model

It is noticeable that the category of normal models are less robust as ε increases. Instead, the adversarial and certified models, which are intrinsically built for that, can support larger image perturbations. The augmentation brought benefits only in the case of the certified contrastive model and slightly improved the performance of the non-contrastive ones.

5.2.2 α, β-CROWN

Since the PGD attack is not a complete method, we decided to perform a more solid verification using α, β-CROWN. The python library alpha-beta-CROWN is used, PGD is disabled and a maximum timeout of 180 seconds for the branch-and-bound phase is set. Based on the robust accuracy previously mentioned, only a subset of models is analyzed. For each of them, 20 image points in which PGD failed are selected and the following results are obtained:

Model	Timeout	Safe	Safe-incomplete
Normal	8	9	3
Normal Contrastive	12	7	1
Adversarial	0	0	20
Adversarial Contrastive	0	0	20
Certified	1	0	19
Certified Contrastive	0	0	20

Table 1: α, β-CROWN results. (Safe-incomplete means that BaB is not needed to verify the input)

From the obtained data, it is only possible to confirm the previous shown trend for the adversarial and certified class models. For the normal ones, instead, the 40% of samples did not complete the verification because of time-out, so no conclusion can be retrieved from them. However, it is still interesting to see that no *unsafe* zones are found. This is clearly due to the steps that BaB performs to get the output. In fact, it branches the initial bounds in sub-bounds and tries to verify the obtained sub-domains. If a zone is not safe, it means that an adversarial example is present in it, so the BaB process needs to perform many steps to find it, many more compared to the *safe* case. This explains the absence of the unsafe status.

6 Final Considerations

Given the data evidence shown in the previous section, the models trained using SPL are in general performing in a worse way with respect to the relative counterpart. Only the normal contrastive model without augmentation and the certified contrastive model with augmentation seems to not respect that trend. Regardless, the first one improves the robustness of 1-2% and the latter one shows a non-admissible level of accuracy to be applied in real world applications. Perhaps scaling the architecture could give more expressive power to the CNN's encoder and the SCL gain could be noticed. This is just an assumption that might be tested in future works.

References

[1] Raia Hadsell, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant

mapping". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 1735–1742.

[2] Prannay Khosla et al. "Supervised Contrastive Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 18661–18673. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

[3] Aaron van den Oord Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).