Masked Relational Knowledge Distillation For Self-Supervised Learning

Cristopher McIntyre-Garcia and Jean-Luc Blais-Amyot
University of Ottawa

Background

Self-supervised learning (SSL) in computer vision is evolving rapidly. Advances have been made, such as implementing the ViT[2], Self-Knowledge Distillation (KD)[3] and performing Masked Image Modelling (MiM)[1] as a pretext task. These techniques have demonstrated the ability to train backbone architectures to generate image representations without the need of labeled data. These backbone models can later be fine-tuned on a smaller set of data and have been shown to perform on par with their supervised backbone counterparts[6]. The framework iBOT[6] (SoTA) combines KD, ViT and MiM all together into one architecture. Relation Knowledge Distillation (RKD)[5] applies the principle of knowledge distillation to a structure representation of images.

Idea

Our proposed idea was to combine ViT, MiM and RKD. To achieve this, we replaced the KD in iBOT with a variation of RKD. This is done by modifying the loss function in the original architecture, which was instance-to-instance cross-entropy. RKD aims to capture the structural representation of similar images. The way to achieve this is by using potential functions such as those proposed in [5]. We aimed to evaluate the properties and performance of each potential function, and their impact on the SSL framework. Our model is seen below.

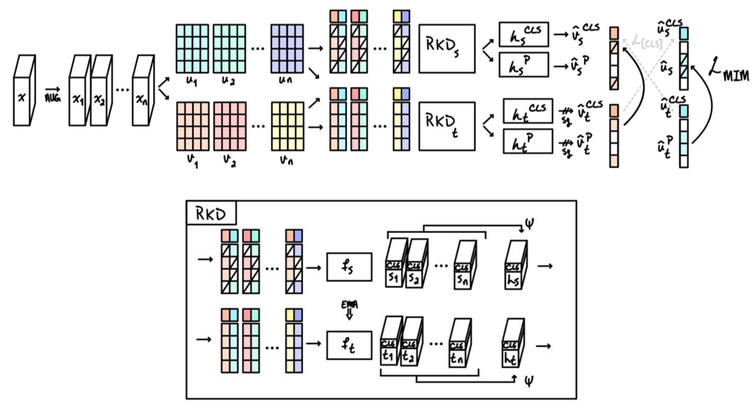

Fig. 1: Relation Knowledge Distillation iBOT

Instead of using labeled data, augmentations are passed through a student-teacher network, and RKD is applied to form the structural components. We figured that using different loss functions would necessitate using different hyperparameters. Therefore, we also had to better our understanding of their impact on the performance.

Loss functions

Two loss functions were proposed in [5]: Distance-wise and Angle-wise loss. Each of them uses the potential metrics 1 and 3 respectively. The purpose behind these functions is to find the fundamental structure of the data, as opposed to only using a point to point comparison.

$$\psi_D(t_i, t_j) = \frac{1}{u} \left\| t_i - t_j \right\|_2 \tag{1}$$

$$\mathcal{L}_{RKD-D} = \sum_{(x_i, x_j) \in X^2} l_\delta \left(\psi_D(t_i, t_j), \psi_D(s_i, s_j) \right) \tag{2}$$

In eq. 1, the euclidean distance between a pair of examples is computed from their output representation over a normalization factor. Eq. 1 is used in a Hubert loss function [4] for each pair of similar images in a batch.

$$\psi_A(t_i, t_j, t_k) = \cos \angle t_i t_j t_k = \left\langle e^{ij}, e^{kj} \right\rangle ; where \;\; e^{ij} = \frac{t_i - t_j}{\left\| t_i - t_j \right\|_2} \tag{3}$$

$$\mathcal{L}_{RKD-A} = \sum_{(x_i, x_j, x_k) \in X^3} l_\delta \left(\psi_A(t_i, t_j, t_k), \psi_A(s_i, s_j, s_k) \right) \tag{4}$$

The angle-wise metric uses a triplet of examples to create an angle. Eq. 3 also originally uses the Huber loss function. Another way of implementing these potential functions is by incorporating them into the loss of iBOT. This way, we compute the class and patch structures and then compute the iBOT loss between them.

Training

In every run, we used a ViT-small/8 as the backbone. The 8 dictates the size of the patches that are fed into the ViT. The dataset used to pre-train the model is a version of CIFAR10 that contains images of size 32x32. 16 patches are therefore fed into the ViT. The batch size used was 256 for pre-training while using regular iBOT and iBOT with RKD-D. When pre-training the ViT using the iBOT with the RKD-A framework, the batch size needed to be lowered to 16 due to virtual memory limitations. Once the backbone model was trained, we could either perform unsupervised classification testing, or we could fine-tune the model for a specific task. In both cases, the testing was done on the CIFAR10 testing set. The fine-tuning and unsupervised training underwent 10 epochs.

Results

The following figure demonstrates the loss and accuracy during training. The accuracy is computed by comparing the output of the teacher with that of the student. The main goal of the framework is to train the student network to accurately predict the same representations as the teacher network, with less information. Once this is achieved, it is assumed that the student network has learned to represent images efficiently. These representations hold malleable information that can be useful for downstream tasks.

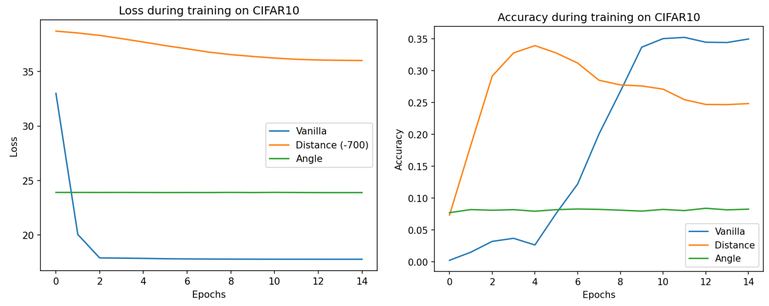

Fig. 2: Training Loss and Accuracy

As shown in the above plots, the loss curves tend to differ largely when using different KD frameworks. When using RKD, the loss starting point seems to be directly proportional to the batch size. Because we were unable to use a large batch size for the RKD-A framework, the loss stayed relatively low. The same cannot be said for the model trained using the RKD-D framework. The loss starting point of the original iBOT framework was also proportional to the batch size, however the impact was not as substantial. The loss of the RKD-D framework lowered at a very slow pace, and converged towards a value after around 12 epochs. The loss of the original iBOT framework dropped too rapidly and flat lines after 2 epochs. The loss of the RKD-A framework did not increase or decrease at all.

The accuracy curves are also quite different. The best short term accuracy is that of the model trained with the RKD-D framework. The model trained with the RKD-A framework does not change, which is representative of its loss curve. The model trained with the regular iBOT framework seems to perform better in the long term. It seems as though a larger loss can be beneficial to the accuracy of a model in the short run. However, due to the fragile nature of SSL frameworks, it is important to note that this advantage most likely would not hold for longer training with larger images.

	Sup Linear		Unsup CLS	
Pre-train	Acc@1	Acc@5	Acc@1	Acc@5
Random init	**33.900**	83.870	32.100	84.240
iBOT & KD	25.650	74.420	30.850	84.110
iBOT & RKD-D	33.820	**84.260**	**32.900**	**84.270**
iBOT & RKD-A	33.720	83.830	31.790	83.860

Table 1. Results on fine-tune linear classification and unsupervised CLS token classification.

The best performing framework was the one using RKD-D. It was able to outperform both the instance KD and the RKD-A frameworks. That said, the performance was not substantially better than using randomly initialized weights. We believe that due to the limited batch size and image size, the models were prone to collapsing. It is unclear whether the framework would outperform the regular iBOT framework when both are trained on larger data. We suspect that the framework would have to undergo more hyperparameter tuning if more training were to be done. We also believe that our implementation has room for improvement.

Attention Please

One interesting property that was discovered while using the ViT as a backbone for SSL networks was their capability of creating attention maps that look like segmented versions of the original image. We attempted to keep this property with our additions to the implementation of iBOT.

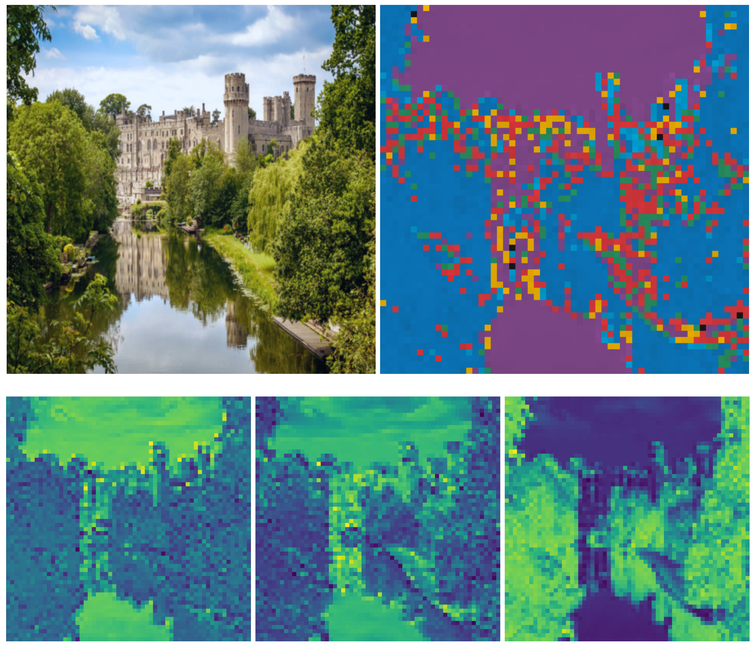

Fig. 3: Attention maps using distance trained model

In the figure above, we see that our model is still able to create these nice segmented maps. Different heads of the ViT are able to segment different objects in the image. When combined, the heads form a semantic segmentation map by separating the respective objects from the image.

Conclusion

The results have shown that pre-training a model using our self-supervised method can be beneficial. With more attention given to the loss functions, we believe better results can be achieved. Though our results are not SoTA, we believe that with better hardware and more fine-tuning of the hyperparameters, we could improve on the works previously conducted.
In future works, we would like to explore different distance metrics or introduce the concept of similarity as a potential metric. We also think that it would be beneficial to attempt different types of knowledge distillation techniques.

Limitations

Computer vision development requires a lot of images and computational power. The machine used to train the models was not as good as the one used to develop iBOT. Because of this, we were restricted to a different dataset and hyperparameters such as batch size, patch size, and training time. We were only able to explore a portion of our ideas due to the time constraints of the project.

References

[1] Hangbo Bao et al. *BEiT: BERT Pre-Training of Image Transformers*. Sept. 2022.
[2] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. May 2021.
[3] Jianping Gou et al. *Knowledge Distillation: A Survey*. May 2021.
[4] Peter J. Huber. "Robust estimation of a location parameter". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101. DOI: 10.1214/aoms/1177703732.
[5] Wonpyo Park et al. *Relational knowledge distillation*. May 2019.
[6] Jinghao Zhou et al. *Ibot: Image Bert pre-training with online tokenizer*. Jan. 2022.