Big Self-Supervised Models are Strong Semi-Supervised Learners

(2020)

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, Geoffrey Hinton

Summary

Contributions

In this paper, the authors investigate a simple improve for semi-supervised learning: task-agnostic pretraining using all of the unlabeled data (labeled can also be optionally used, but for small set, the gains are small) with contrastive learning, then fine tuning using the labeled data. And show that even if the first step is task agnostic, the obtained results are better that the majority of SSL methods, and that the smaller the labeled set is, the more we can benefit from a bigger pretrained model. Such a big pretrained model can then be distilled into a smaller network for efficient usage, when the fine-tunned big model, can be distilled in a task-specific way to a smaller model without losing any performance.

So to summarize, the model is first trained with a contrastive loss, i.e., to learn embeddings where similar images, i.e. positives, where a given image and its transformed version need to have similar features, while other negatives, i.e., randomly sampled images, are different than the feature of the positives. Then, a fine tuning stage of the backbone (i.e., the projection head used to go from high dimensional space, like 2048 to a lower embeddings space for contrastive learning, like 128, is removed), then the fine tunned model is distilled for an efficient usage using standard distillation loss, with soft labels in a cross-entropy loss produced by the teacher, which is the big model.

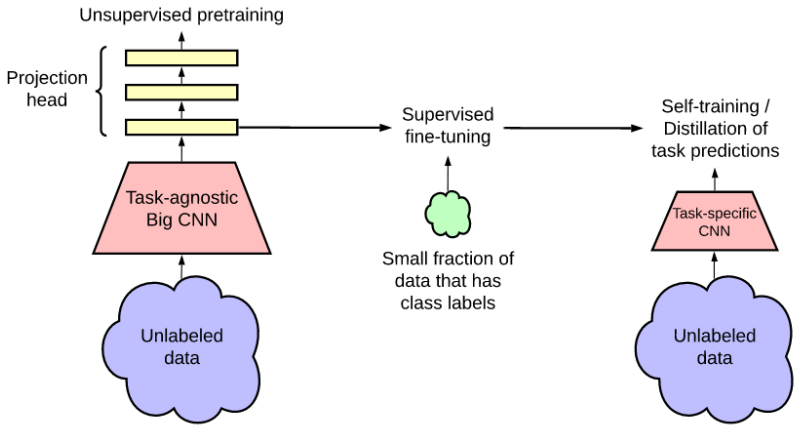


Figure 3: The proposed semi-supervised learning framework leverages unlabeled data in two ways: (1) task-agnostic use in unsupervised pretraining, and (2) task-specific use in self-training / distillation.

Method

Stage 1: Contrastive learning The first stage, consisting of self-supervised pretraining using a contrastive loss, where given a randomly sampled batch of image, each image x_i is augmented twice with random croping, color distortion and blur. Resulting in two positives: x_{2k-1} and x_{2k}, which are

the encoded by a backbone $f(.)$, then projected into the embedding space using a projection head $g(.)$, ending up with two features z_{2k-1} and z_{2k}, that we need to pull together in the embedding space, and push other randomly sampled negative away from them, and this is done using a contrastive loss

$$\ell_{i,j}^{\mathrm{NT-Xent}} = -\log \frac{\exp\left(\mathrm{sim}\left(z_i, z_j\right)/\tau\right)}{\sum_{k=1}^{2N} 1_{[k\neq i]} \exp\left(\mathrm{sim}\left(z_i, z_k\right)/\tau\right)}$$

with cosine similarity as the similarity measure between the embeddings $\mathrm{sim}(.,.)$. The authors also proposed some modification to existing self-supervised pipelines for better pre-training, such adjustments consists of: (1) Using deeper but less wide networks with a channel wise attention mechanism (selective kernel). (2) A deeper projection head $g(.)$, and in the fine-tuning stage, some layers are re-used. (2) Using a queue of embeddings, where and exponential moving average of the network is used to generate enqueued embeddings, to be used as negatives in the contrastive loss.

Stage 2: Fine tuning Given a 3 layer projection head: $g(\boldsymbol{h}_i) = W^{(3)}(\sigma(W^{(2)}\sigma(W^{(1)}\boldsymbol{h}_i))$, the task specific model f^{task} is fine tuning by adding a liner layer W^{task} to project from a given space to the output classes. In this case, we can either use some layer of the projection head as input: $f^{\text{task}}\left(\boldsymbol{x}_i\right) = W^{\text{task}}\sigma\left(W^{(1)}f\left(\boldsymbol{x}_i\right)\right)$, or only the backbone: $f^{\text{task}}\left(\boldsymbol{x}_i\right) = W^{\text{task}}f\left(\boldsymbol{x}_i\right)$

Stage 3: Distillation To reduce the size of the fine tunned model, which is usually big, and to further improve the performance, the authors conduct a last step when the all of the data is used a self-training manner using the big model to train a smaller one. The teacher, here the bigger and fine tunned model, predict class probabilities on the downstream task on the unlabeled examples, that are considered as pseudo-labels and used in a standard cross-entropy loss for distillation, in addition to the normal supervised loss over labeled examples.

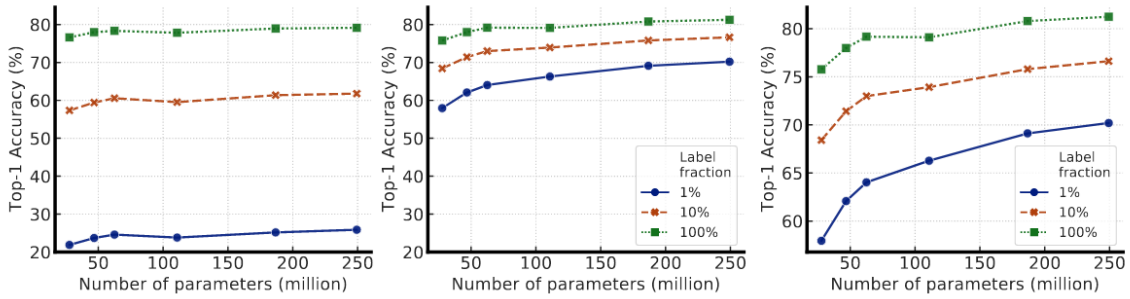
$$\mathcal{L} = -(1-\alpha)\sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}^L}\left[\log P^S\left(y_i \mid \boldsymbol{x}_i\right)\right] - \alpha\sum_{\boldsymbol{x}_i\in\mathcal{D}}\left[\sum_y P^T\left(y \mid \boldsymbol{x}_i;\tau\right)\log P^S\left(y \mid \boldsymbol{x}_i;\tau\right)\right]$$

where $P\left(y \mid \boldsymbol{x}_i\right) = \exp\left(f^{\text{task}}\left(\boldsymbol{x}_i\right)[y]/\tau\right) / \sum_{y'}\exp\left(f^{\text{task}}\left(\boldsymbol{x}_i\right)[y']/\tau\right)$ and τ is the temperature scalar.

Results

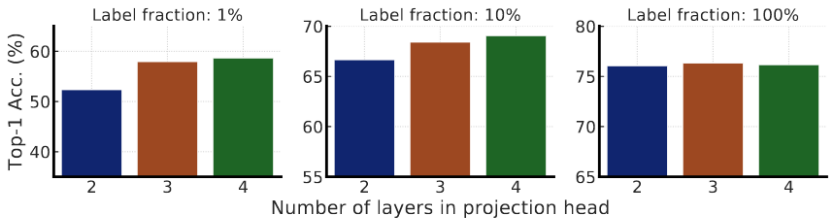
Table 1: Top-1 accuracy of fine-tuning SimCLRv2 (on varied label fractions) or training a linear classifier on the ResNet output. The supervised baselines are trained from scratch using all labels in 90 epochs. The parameter count only include ResNet up to final average pooling layer. For fine-tuning results with 1% and 10% labeled examples, the models include additional non-linear projection layers, which incurs additional parameter count (4M for 1× models, and 17M for 2× models). See Table G.1 for Top-5 accuracy.

Depth	Width	SK	Param (M)	F-T (1%)	F-T (10%)	F-T (100%)	Linear eval	Supervised
50	1×	False	**24**	**57.9**	**68.4**	**76.3**	**71.7**	**76.6**
		True	35	64.5	72.1	78.7	74.6	78.5
	2×	False	94	66.3	73.9	79.1	75.6	77.8
		True	140	70.6	77.0	81.3	77.7	79.3
101	1×	False	43	62.1	71.4	78.2	73.6	78.0
		True	65	68.3	75.1	80.6	76.3	79.6
	2×	False	170	69.1	75.8	80.7	77.0	78.9
		True	257	73.2	78.8	82.4	79.0	80.1
152	1×	False	58	64.0	73.0	79.3	74.5	78.3
		True	89	70.0	76.5	81.3	77.2	79.9
	2×	False	233	70.2	76.6	81.1	77.4	79.1
		True	354	74.2	79.4	82.9	79.4	80.4
152	3×	True	**795**	**74.9**	**80.1**	**83.1**	**79.8**	**80.5**

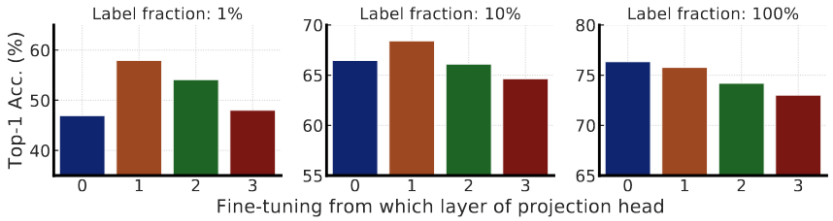


(a) Supervised (b) Semi-supervised (c) Semi-supervised (y-axis zoomed)

Figure 4: Top-1 accuracy for supervised vs semi-supervised (SimCLRv2 fine-tuned) models of varied sizes on different label fractions. ResNets with depths of 50, 101, 152, width multiplier of $1\times$, $2\times$ (w/o SK) are presented here. For supervised models on 1%/10% labels, AutoAugment [34] and label smoothing [35] are used. Increasing the size of SimCLRv2 models by $10\times$, from ResNet-50 to ResNet-152 ($2\times$), improves label efficiency by $10\times$.



(a) Effect of projection head's depth when fine-tuning from optimal middle layer.

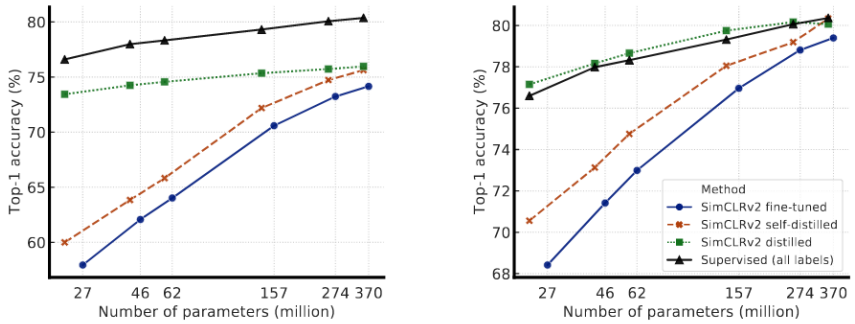


(b) Effect of fine-tuning from middle of a 3-layer projection head (0 is SimCLR).

Figure 5: Top-1 accuracy via fine-tuning under different projection head settings and label fractions (using ResNet-50).

Table 2: Top-1 accuracy of a ResNet-50 trained on different types of targets. For distillation, the teacher is ResNet-50 ($2\times$+SK), and the temperature is set to 1.0. The distillation loss (Eq. 2) does not use label information. Neither strong augmentation nor extra regularization are used.

Method	Label fraction	
	1%	10%
Label only	12.3	52.0
Label + distillation loss (on labeled set)	23.6	66.2
Label + distillation loss (on labeled+unlabeled sets)	69.0	75.1
Distillation loss (on labeled+unlabeled sets; our default)	68.9	74.3



(a) Label fraction 1%

(b) Label fraction 10%

Figure 6: Top-1 accuracy of distilled SimCLRv2 models compared to the fine-tuned models as well as supervised learning with all labels. The self-distilled student has the same ResNet as the teacher (without MLP projection head). The distilled student is trained using the self-distilled ResNet-152 (2×+SK) model, which is the largest model included in this figure.