Generative Pretraining from Pixels
(2020)

Mark Chen et al.

Summary

Contributions

The authors in this paper investigate the effectiveness of using a pretraining objectives with transformer, similar to the ones used in NLP, where a model is first trained on language modeling. Either in an autoregressive manner, predicting future tokens conditioned on the past ones, or with a masking objective, predicting the masked tokens. But his time applied to pixels, where we either predict masked pixel or futures ones. Then analyze the quality of the learned features ether by linear probing, adding a liner layer over the learned and frozen features, or fine-tuning, where the all of the model's parameters are fine-tunned in addition to the added classification layer.

So the method consist of the following steps: (1) reducing the dimensionality of the images, since the model used is based on transformers, with quadratic complexity, the images are first down-sampled and the colors are also converted to a representation fewer bits, then (2) pretraining stage, with either an autoregressive or masking (i.e., BERT) objective, (3) finally, using the pretrained model on the downstream task, with either linear-probing or fine-tuning.

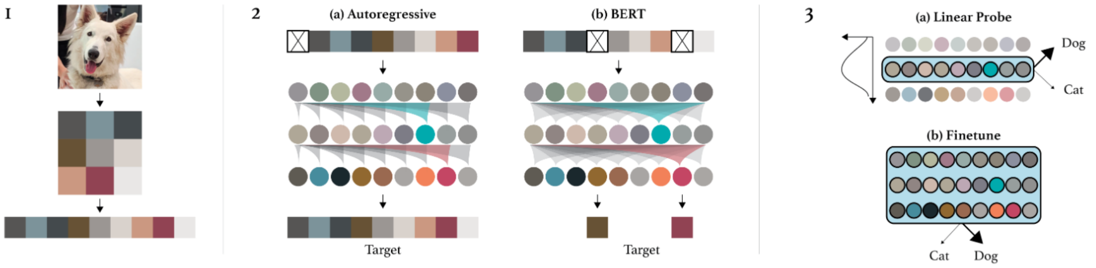

Figure 1. An overview of our approach. First, we pre-process raw images by resizing to a low resolution and reshaping into a 1D sequence. We then chose one of two pre-training objectives, auto-regressive next pixel prediction or masked pixel prediction. Finally, we evaluate the representations learned by these objectives with linear probes or fine-tuning.

Method

Step 1: Pre-processing The paper uses lightweight data augmentation: random crops, resizing, padding with pixel reflection and H flips, depending on the dataset. Then the dimensionality of the input is reduced: First, the image are resized to: $32^2 \times 3$, $48^2 \times 3$ or $64^2 \times 3$. However, the three colors channels still makes the input dimensionality quite big. The images are converted to a new 9-bit color palette by clustering RGB values using k-means with $k = 512$. Further reducing the input dimensionality by a factor of 3.

Step 2: Pre-training Given an unlabeled dataset X consisting of high dimensional data x, the model is trained to model the density $p(x)$ in an auto-regressive manner with a given permutation π:

$$p(x) = \prod_{i=1}^{n} p\left(x_{\pi_i} \mid x_{\pi_1}, \ldots, x_{\pi_{i-1}}, \theta\right)$$

Such an objective can be optimized using two possible losses, and autoregressive loss (AR) or a masking objective (BERT), where some M tokens of the input are masked, and the model is trained to predict them:

$$L_{AR} = \underset{x\sim X}{\mathbb{E}}[-\log p(x)]$$

$$L_{BERT} = \underset{x\sim X}{\mathbb{E}}\,\underset{M}{\mathbb{E}}\sum_{i\in M}\left[-\log p\left(x_i \mid x_{[1,n]\setminus M}\right)\right]$$

Step 3: Fine-tuning When fine-tuning, the output of the transformer model is a sequence of $d-$dimensional vector, to get the representation of the whole image, we first average pool over the sequence dimension resulting in an image level representation f^L. Which is then used as input to the classifier. Trained using only the cross entropy loss when doing probing, or both losses when doing fine-tuning.

Results

Table 1. Comparing linear probe accuracies between our models and state-of-the-art models utilizing unsupervised ImageNet transfer or supervised ImageNet transfer.

Model	Acc	Unsup Transfer	Sup Transfer
CIFAR-10			
ResNet-152	94		√
SimCLR	95.3	√	
iGPT-L	96.3	√	
CIFAR-100			
ResNet-152	78.0		√
SimCLR	80.2	√	
iGPT-L	82.8	√	
STL-10			
AMDIM-L	94.2	√	
iGPT-L	95.5	√	

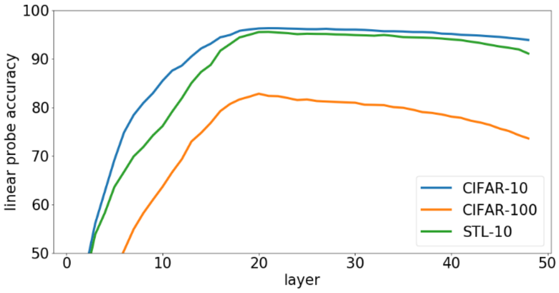

Figure 2. Representation quality depends on the layer from which we extract features. In contrast with supervised models, the best representations for these generative models lie in the middle of the network. We plot this unimodal dependence on depth by showing linear probes for iGPT-L on CIFAR-10, CIFAR-100, and STL-10.

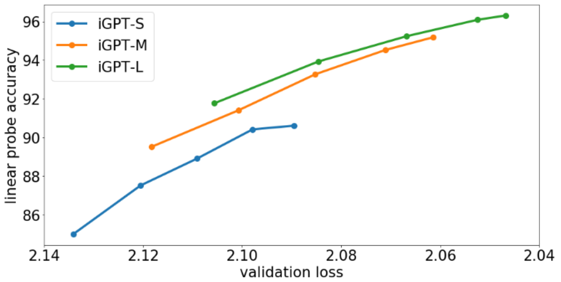

Figure 3. Plot of representation quality as a function of validation generative loss. Each line tracks a model throughout generative pre-training: the dotted markers denote checkpoints at steps 65K, 131K, 262K, 524K, and 1000K. The positive slope suggests a link between improved generative performance and improved representation quality. Larger models produce better representations than smaller ones both at the end of training and at the same value of validation loss. iGPT-XL is not shown since it was trained on a different dataset.

Table 2. Comparing linear probe accuracies between our models and state-of-the-art self-supervised models. A blank input resolution (IR) corresponds to a model working at standard ImageNet resolution. We report the best performing configuration for each contrastive method, finding that our models achieve comparable performance.

Method	IR	Params (M)	Features	Acc
Rotation	orig.	86	8192	55.4
iGPT-L	$32^2 \cdot 3$	1362	1536	60.3
BigBiGAN	orig.	86	8192	61.3
iGPT-L	$48^2 \cdot 3$	1362	1536	65.2
AMDIM	orig.	626	8192	68.1
MoCo	orig.	375	8192	68.6
iGPT-XL	$64^2 \cdot 3$	6801	3072	68.7
SimCLR	orig.	24	2048	69.3
CPC v2	orig.	303	8192	71.5
iGPT-XL	$64^2 \cdot 3$	6801	15360	72.0
SimCLR	orig.	375	8192	76.5

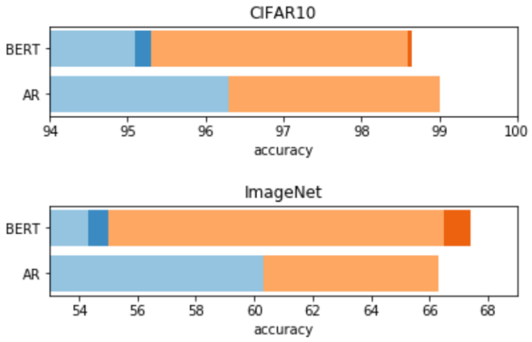

Figure 4. Comparison of auto-regressive pre-training with BERT pre-training using iGPT-L at an input resolution of $32^2 \times 3$. Blue bars display linear probe accuracy and orange bars display fine-tune accuracy. Bold colors show the performance boost from ensembling BERT masks. We see that auto-regressive models produce much better features than BERT models after pre-training, but BERT models catch up after fine-tuning.

Table 3. Comparing fine-tuning performance between our models and state-of-the-art models utilizing supervised ImageNet transfer. We also include AutoAugment, the best performing model trained end-to-end on CIFAR. Table results: AutoAugment (Cubuk et al., 2019), SimCLR (Chen et al., 2020), GPipe (Huang et al., 2019), EfficentNet (Tan & Le, 2019)

Model	Acc	Unsup Transfer	Sup Transfer
CIFAR-10			
AutoAugment	98.5		
SimCLR	98.6	✓	
GPipe	99.0		✓
iGPT-L	99.0	✓	
CIFAR-100			
iGPT-L	88.5	✓	
SimCLR	89.0	✓	
AutoAugment	89.3		
EfficientNet	91.7		✓

Table 4. Comparing performance on low-data CIFAR-10. By leveraging many unlabeled ImageNet images, iGPT-L is able to outperform methods such as Mean Teacher (Tarvainen & Valpola, 2017) and MixMatch (Berthelot et al., 2019) but still underperforms the state of the art methods (Xie et al., 2019; Sohn et al., 2020). Our approach to semi-supervised learning is very simple since we only fit a logistic regression classifier on iGPT-L's features without any data augmentation or fine-tuning - a significant difference from specially designed semi-supervised approaches. Other results reported from FixMatch (Sohn et al., 2020).

Model	40 labels	250 labels	4000 labels
Mean Teacher		32.3 ± 2.3	9.2 ± 0.2
MixMatch	47.5 ± 11.5	11.0 ± 0.9	6.4 ± 0.1
iGPT-L	26.8 ± 1.5	12.4 ± 0.6	5.7 ± 0.1
UDA	29.0 ± 5.9	8.8 ± 1.1	4.9 ± 0.2
FixMatch RA	13.8 ± 3.4	5.1 ± 0.7	4.3 ± 0.1
FixMatch CTA	11.4 ± 3.4	5.1 ± 0.3	4.3 ± 0.2