Image GPT

Sangho Lee

Paper Understanding

Abstract

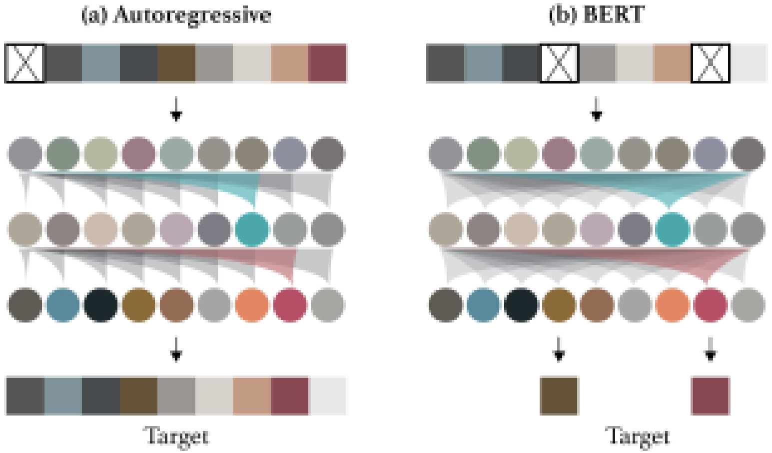
- Image GPT from OpenAI
- GPT?
  - Language model pretrained to predict next word with decoders from 40GB web-crawled text
  - *I am a* _______
- Motivation
  - Inspired by progress in unsupervised representation learning for natural language
  - Can learn useful representation from images
- Train a sequence Transformer to auto-regressively predict pixels
  - NLP words → Image pixels

Approach

- Consists of a pre-training stage and fine-tuning stage
- Pre-training
  - Apply the sequence Transformer architecture to predict pixels instead of language tokens
  - Auto-regressive, BERT

$$p(x) = \prod_{i=1}^{n} p(x_{\pi_i} | x_{\pi_1}, ..., x_{\pi_{i-1}}, \theta)$$

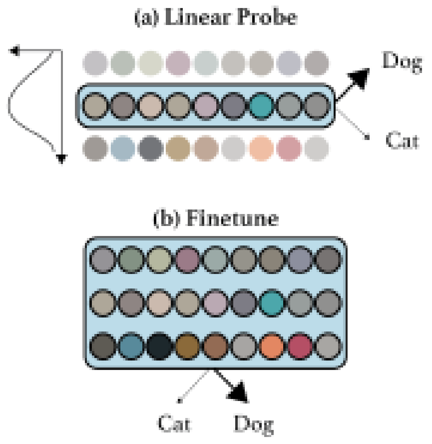
$$L_{AR} = \mathop{\mathbb{E}}_{x \sim X} [-\log p(x)]$$



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

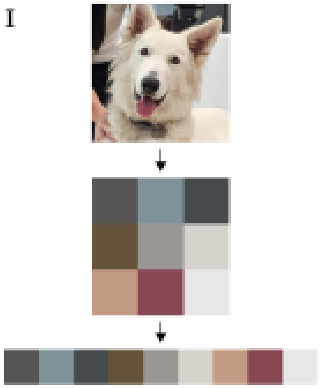
Approach

- Fine-tuning & Linear probing
  - Linear probe
    - Generate class logit with features from medium layer
  - Fine-tuning
    - Extract class logit from sequence dimension by average polling



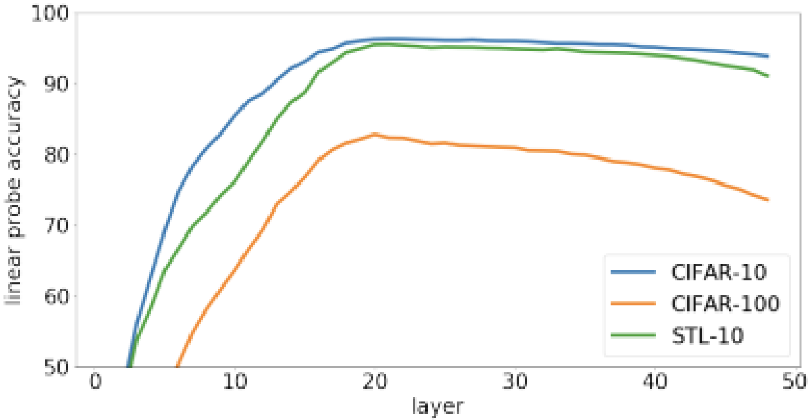
Methodology

- Data augmentation
  - No augmentation when pre-training except randomly 224 x 224 crop
  - For full-network fine-tuning, pad 4 pixels on each side and randomly 32 x 32 crop / horizontal flip
- Context reduction
  - Memory requirements of the transformer decoder increase quadratically
  - (224x224x3 → Tens of thousands of larger than language models)
  - Resize image to low resolution (32x32x3, 48x48x4, 64x64x4)
  - RGB to clusters using k-means → 32x32, 48x48, 64x64
- Model
  - iGPT-XL (60 layers, d=3072, 6.8B parameters)
  - iGPT-L (48 layers, d=1536)
  - iGPT-S (24 layers, d=512)



Experiments

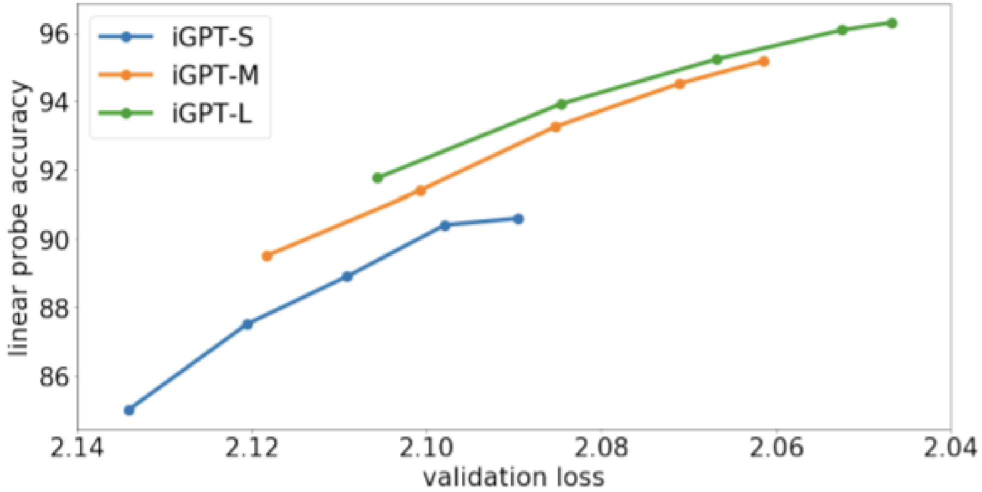
- What representation works best in a generative model without latent variables
  - In supervised pre-training, second to last layer tends to have best quality representation
  - To find representation works best from generative pre-training, test with each layers



  - Representations improve as a function of depth, starting around the middle layer, begin to deteriorate

Experiments

- Better generative models learn better representations
  - With high validation performance and high capacity shows better representations in linear probing



Experiments

- On CIFAR and STL-10
  - Best performance when pretrained on ImageNet and classify on CIFAR and STL-10

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	√	

Experiments

- On ImageNet
  - High accuracy(?) trained with row resolution image

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

Contributions & Limitations

- Contributions
  - Suggests generative image modeling continues to be a promising route to learn high-quality unsupervised image representations
  - Simply predicting pixels learns state of the art representations for low resolution datasets

- Limitations
  - Currently model low-resolution inputs
  - Requires large models to learn high quality representations (iGPT-L has 2 to 3 times as many parameters as similarly performing models on ImageNet and uses more compute)

Model Implementation & Test

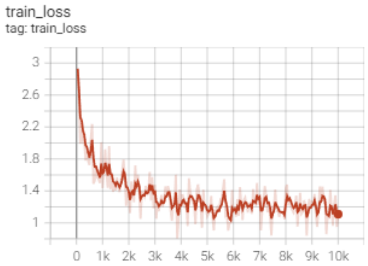
Environments

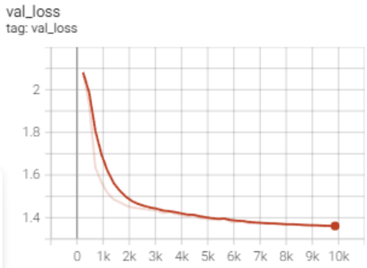
- Environments
  - 3 A100 GPU, CUDA 11.2
  - Framework: Pytorch Lightning

- Code description
  - run.py : Execution code
  - data.py : Data preprocessing
  - gpt2.py : GPT2 model
  - image_gpt.py : PL module for run image GPT in Pytorch Lightning

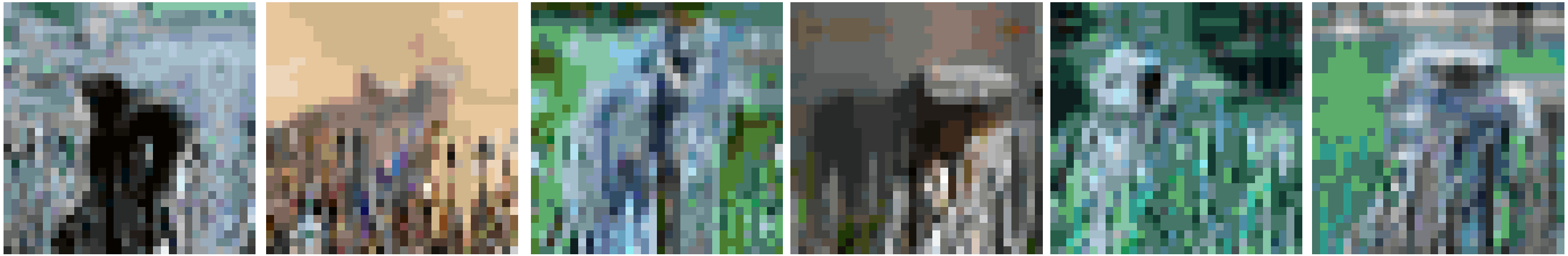
Test Result

- Loss (After 50 epochs)
  - Train loss : 1.092
  - Validation loss : 1.36





- FID-5K : 6.724 (feature 64), 409.865 (feature 2048)
- Selected samples



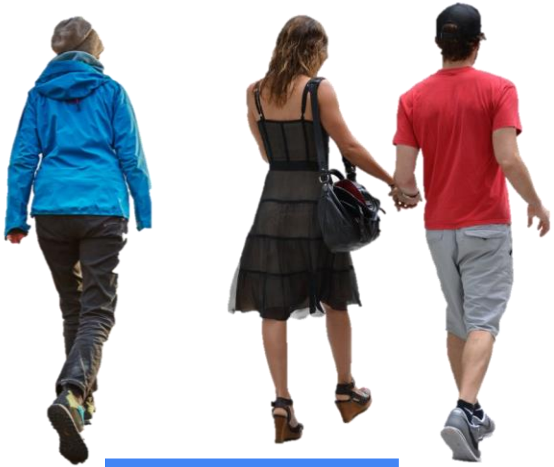
Next Research Plan

Research Plan

- Topic : Generate image with specific pose or action
- Method
  - Pretrain pose or action representation from weakly labelled dataset with image-GPT
  - Generate images with target pose/action and style
- Dataset plan

Side

Front



Backward

Research Plan

- Candidate contribution
  - Can be used without specific pose dataset regardless target objects
    - Usually pose transfer / generation model uses pretrained pose network to extract pose
    - No such dataset or pretrained dataset except human (animals, animation, etc.)
  - Can generate pose or action without reality
    - If we need image with unreality, we can just draw and add any label to such images