Generative Adversarial Text to Image

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee

Presenter: Ryan Wolter
Motivation

- **Goal** - Generate plausible images from a single text sentence
- **Why?**
  - Generative models could replace (expensive) humans in content creation pipeline
    - HOWEVER, these models must be conditioned on an input given by an operator to be useful.
  - Guided style transfer / image manipulation
    - Use natural language as the interface for manipulating images
  - It's an unsolved problem...
    - ...Because it’s hard.
      - The mapping from text to image has numerous “correct” answers
Key Contributions

● First “end-to-end fully differentiable architecture” for text to image
  ○ Whole model is a GAN, not just using a GAN for post processing

● Conditioning off text
  ○ Other conditional generative models key off “attributes” or “class labels”

● Use of interpolation of text features to aid in stabilizing the GAN
Why now?

- RNNs for creating discriminative text features
- CNNs for image features
- GANs (Generative Adversarial Networks) for image generation
So what are GANs?

- “GANs are an exciting new…” - any paper that references (Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.)

Discriminator: MAX -> $E[ \log(D(X))] + E[ \log(1-D(G(Z)))]$

Generator: MIN -> $E[ \log(1-D(G(Z)))]$

Fake = 0, True = 1
How do we condition on text?

- Two tasks
  - Learn a text feature representation of sentence
  - Then generate an image from that representation
How do we condition on text?

- Two tasks
  - Learn a text feature representation of sentence
  - Then generate an image from that representation
Deep symmetric structured joint embedding

- Or, Learning a text feature representation of sentence

Loss:

\[
\frac{1}{N} \sum_{n=1}^{N} \Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n))
\]

Vision Classifier:

\[
f_v(v) = \arg \max_{y \in Y} E_{v \sim \mathcal{V}(y)} [\phi(v)^T \varphi(t)]
\]

Text Classifier:

\[
f_t(t) = \arg \max_{y \in Y} E_{v \sim \mathcal{V}(y)} [\phi(v)^T \varphi(t)]
\]
Deep symmetric structured joint embedding

- Or, Learning a text feature representation of sentence

Figure 1: Our model learns a scoring function between images and text descriptions. A word-based LSTM is shown here, but we also evaluate several alternative models.
How do we condition on text?

- Two tasks
  - Learn a text feature representation of sentence
  - Then generate an image from that representation
Condition on Text

\[ X, t \]
(Real Images, Matching Text)

\[ G(Z, t) \]
(Fake Images, Conditioning Text)

A wizard with a long beard and a staff

A wizard with a long beard and a staff

Discriminator: \( \text{MAX} \rightarrow E[ \log(D(X, t)) ] + E[ \log(1-D(G(Z, t))) ] \)
Condition on Text

- We need to not only say “Does this image look right”, but also, “and does it match the text?"
- Discriminator now provides feedback for three scenarios instead of two

\[ X, t \]
(Real Images, Matching Text)

\[ G(Z, t) \]
(Fake Images, Conditioning Text)

A wizard with a long beard and a staff

A wizard with a long beard and a staff

**Discriminator: MAX ->** \[ E[ \log(D(X, t))] + E[ \log(1-D(G(Z, t))) ] \]
Condition on Text

- We need to not only say “Does this image look right”, but also, “and does it match the text?”
- Discriminator now provides feedback for three scenarios instead of two

\[
\begin{align*}
  X, t \quad \text{(Real Images, Matching Text)} \\
  G(Z, t) \quad \text{(Fake Images, Conditioning Text)} \\
  X, t' \quad \text{(Real Images, Mismatched Text)}
\end{align*}
\]

A wizard with a long beard and a staff

A wizard with a long beard and a staff

An elf with pointy ears and a bow.

Discriminator: \( \text{MAX} \rightarrow \mathbb{E} [ \log(D(X, t)) ] + \mathbb{E} [ \log(1-D(G(Z, t))) ] + \mathbb{E} [ \log(1-D(X,t')) ] \)
Architecture

- Based on Deconvolutional GANs
  - Simply just convolution layers in reverse
- Run the text through a text encoder to create a vector (size 128 in the paper)
- Concatenate this encoded vector to the noise, $z$, for the generator
- Concatenate this encoded vector to a middle layer of the discriminator
- Add the additional loss as described previously for the real image with mismatched text $(X, t')$
This flower has small, round violet petals with a dark purple center

$\varphi(t) \rightarrow \varphi \rightarrow \hat{x} := G(z, \varphi(t))$

This flower has small, round violet petals with a dark purple center

$D(\hat{x}, \varphi(t))$

**Figure 2.** Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.
Algorithm 1 GAN-CLS training algorithm with step size $\alpha$, using minibatch SGD for simplicity.

1: **Input**: minibatch images $x$, matching text $t$, mis-matching $\hat{t}$, number of training batch steps $S$
2: **for** $n = 1$ **to** $S$ **do**
3:     $h \leftarrow \varphi(t)$ \{Encode matching text description\}
4:     $\hat{h} \leftarrow \varphi(\hat{t})$ \{Encode mis-matching text description\}
5:     $z \sim \mathcal{N}(0, 1)^2$ \{Draw sample of random noise\}
6:     $\hat{x} \leftarrow G(z, h)$ \{Forward through generator\}
7:     $s_r \leftarrow D(x, h)$ \{real image, right text\}
8:     $s_w \leftarrow D(x, \hat{h})$ \{real image, wrong text\}
9:     $s_f \leftarrow D(\hat{x}, h)$ \{fake image, right text\}
10: $\mathcal{L}_D \leftarrow \frac{\log(s_r) + (\log(1 - s_w) + \log(1 - s_f))}{2}$
11: $D \leftarrow D - \alpha \frac{\partial \mathcal{L}_D}{\partial D}$ \{Update discriminator\}
12: $\mathcal{L}_G \leftarrow \log(s_f)$
13: $G \leftarrow G - \alpha \frac{\partial \mathcal{L}_G}{\partial G}$ \{Update generator\}
14: **end for**
GAN-CLS-INT

- Variation on GAN-CLS
- The text embedding space is just a real valued vector
- We can interpolate between different vectors to enhance the dataset, and fill the gaps in the generator

\[
\mathbb{E}_{t_1, t_2 \sim p_{data}} \left[ \log(1 - D(G(z, \beta t_1 + (1 - \beta)t_2))) \right]
\]

- This goal is added to the generator
- The discriminator learns to discriminate the pairs without additional labeling, doesn’t need to worry about this part.
Datasets

- **Caltech-UCSD Birds (CUB)**
  - 11,788 Images of birds of 200 classes
  - 5 Captions per image

- **Oxford Flowers**
  - 8,189 Images of flowers from 102 classes
  - 5 Captions per image

- **MS COCO**
  - 100k+ images, each with 5 captions
Results

Caltech-UCSD Birds

GAN-INT - interpolation needed to get plausible images.
Results

Oxford-102 Flowers produces better results, likely because birds have significantly more structure to them.
Results - COCO

GT | Ours
---|---
a group of people on skis stand on the snow.
a table with many plates of food and drinks
two giraffe standing next to each other in a forest.
a large blue octopus kite flies above the people having fun at the beach.
a man in a wet suit riding a surfboard on a wave.
two plates of food that include beans, guacamole and rice.
a green plant that is growing out of the ground.
there is only one horse in the grassy field.
Examining Style Transfer

- There are two inputs in generation: text conditioning, and the random noise, $z$
  - How much does each input contribute to final image?
  - Pose information is stored in some of the noise space ($z$)

---

**Text descriptions (content) | Images (style)**

- The bird has a **yellow breast** with grey features and a small beak.
- This is a large **white bird** with **black wings** and a **red head**.
- A small bird with a **black head** and **wings** and features grey wings.
- This bird has a **white breast**, brown and white coloring on its head and wings, and a thin pointy beak.
- A small bird with **white base** and **black stripes** throughout its belly, head, and feathers.

Different styles ($z$) from existing images
Examining Style Transfer

- Interpolation between encoded text vectors with LS kept constant.
Future Work

● Higher Resolution
  ○ Use de-bluring/upscaling GANS in second stage
  ○ StackGAN https://arxiv.org/abs/1612.03242
  ○ AttenGAN https://arxiv.org/abs/1711.10485
  ○ Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network
StackGAN

This bird is white with some black on its head and wings, and has a long orange beak.

This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face.

This flower has overlapping pink pointed petals surrounding a ring of short yellow filaments.

(a) StackGAN
Stage-I
64x64 images

(b) StackGAN
Stage-II
256x256 images

(c) Vanilla GAN
256x256 images
this bird is red with white and has a very short beak
Photographic Text-to-Image Synthesis with a Hierarchically nested Adversarial Network
Photographic Text-to-Image Synthesis with a Hierarchically nested Adversarial Network

This is a small bird with its body covered in blue feathers, and some brown feathers on its wings.

A small bird with brown wings, vanilla break and a small black beek.
Future Work

- **Higher Resolution**
  - Use de-blurring/upscaling GANS in second stage

- **More datasets**
  - Higher variety images?
  - Long strings of text (i.e. summaries or paragraphs)

- **Establishing metrics to determine “fit” of image to text**
  - How well is information about the text used?
    - The paper make it look like this part just works, but I’m skeptical

- **Incorporating Pre-trained LMs as text encoder**
  - ELMO / BERT / OpenAI GP2
Critique

● (-) Evaluation
  ○ Only qualitative
  ○ No human evaluations

● (-) “Inverting Generator for Style Transfer”
  ○ This isn’t well motivated when introduced, making it hard to see its purpose

● (+) Examining style transfer useful to explain effect of $z$ and conditioning text

● (-) Motivation
  ○ Authors don’t provide any motivation themselves for the task beyond saying it’s hard
Critique (Continued)

- (-) What is GAN-CLS?
  - They never provide a name for the acronym
  - I assume it means “Conditioned with Latent Space? Or something..”
- (+) They provide their model (in Torch) with pre-trained weights!
- (-) No discussion at all on training times and hardware
- (+) Very simple model, end-to-end trainable
- (+) The provided graphics for architecture for the model, as well as the algorithm is very well created, properly labeled, and easy to understand.
Discussion...