10.8工作小结：

1. Paper：Generative Adversarial Text to Image Synthesis

摘要：本文将文本和图像练习起来，根据文本生成图像，结合 CNN 和 GAN 来有效的进行无监督学习。

Attribute Representation: 是一个非常具有意思的方向。由图像到文本，可以看做是一个识别问题；从文本到图像，则不是那么简单。

因为需要解决这两个小问题：

1. learning a text feature representation that captures the important visual deatails ;

2. use these features to synthesize a compelling image that a human might mistake for real.

幸运的是，深度学习对这两个问题都有了较好的解决方案，即：**自然语言表示** 和 **image synthesis** 。

但是，**仍然存在的一个问题**是：**the distribution of images conditioned on a text description is highly multimodal，in the sense that there are very many plausible configurations of pixels that correctly illustrate the description.**

The text classifier induced by the learned correspondence function f_t is trained by optimizing the following structured loss:

$$\frac{1}{N}\sum_{n=1}^{N}\Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n)) \tag{2}$$

where $\{(v_n, t_n, y_n) : n = 1, ..., N\}$ is the training data set, Δ is the 0-1 loss, v_n are the images, t_n are the corresponding text descriptions, and y_n are the class labels. Classifiers f_v and f_t are parametrized as follows:

$$f_v(v) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_{t\sim\mathcal{T}(y)}[\phi(v)^T\varphi(t))] \tag{3}$$

$$f_t(t) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_{v\sim\mathcal{V}(y)}[\phi(v)^T\varphi(t))] \tag{4}$$

where ϕ is the image encoder (e.g. a deep convolutional neural network), φ is the text encoder (e.g. a character-level CNN or LSTM), $\mathcal{T}(y)$ is the set of text descriptions of class y and likewise $\mathcal{V}(y)$ for images. The intuition here is that a text encoding should have a higher compatibility score with images of the correspondong class compared to any other class and vice-versa.

4. Method

Our approach is to train a deep convolutional generative adversarial network (DC-GAN) conditioned on text features encoded by a hybrid character-level convolutional-recurrent neural network. Both the generator network G and the discriminator network D perform feed-forward inference conditioned on the text feature.

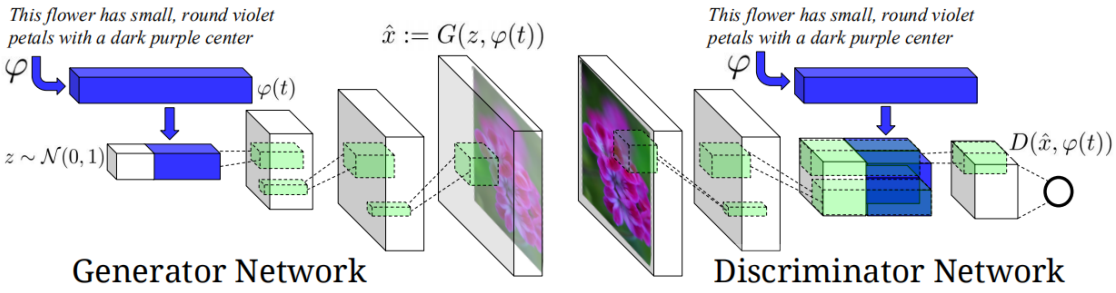


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

文本被经过一个全连接后连接到noise上输入G中进行运算

In naive GAN, the discriminator observes two kinds of inputs: real images with matching text, and synthetic images with arbitrary text. Therefore, it must implicitly separate two sources of error: unrealistic images (for *any* text), and

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

在本文中，图像分类器用的是GoogLeNet，文本分类器用的是LSTM和CNN。

得到文本特征后，需要把文本特征压缩后与图像特征拼接在一起，放入DC-GAN。

随后本文还提出了一个差值学习**4.3. Learning with manifold interpolation (GAN-INT)**

.基于这个属性，我们可以通过简单地在训练集标题的嵌入之间进行插值来生成大量额外的文本嵌入。重要的是，这些插值的文本嵌入不需要对应于任何实际的人工书写的文本，因此没有额外的标签成本。这可以看作是在生成器的目标中增加了一个额外的术语，以最小化：

$$\mathbb{E}_{t_1,t_2\sim p_{data}}[\log(1-D(G(z,\beta t_1+(1-\beta)t_2)))] \quad (5)$$

where z is drawn from the noise distribution and β interpolates between text embeddings t_1 and t_2. In practice we found that fixing $\beta = 0.5$ works well.

Because the interpolated embeddings are synthetic, the discriminator D does not have "real" corresponding image and text pairs to train on. However, D learns to predict whether image and text pairs match or not. Thus, if D does a good job at this, then by satisfying D on interpolated text embeddings G can learn to fill in gaps on the data manifold in between training points. Note that t_1 and t_2 may come from different images and even different categories.[1]

插值法的妙用\Uparrow$

4.4. Inverting the generator for style transfer

If the text encoding $\varphi(t)$ captures the image content (e.g. flower shape and colors), then in order to generate a realistic image the noise sample z should capture style factors such as background color and pose. With a trained GAN, one may wish to transfer the style of a query image onto the content of a particular text description. To achieve this, one can train a convolutional network to invert G to regress from samples $\hat{x} \leftarrow G(z, \varphi(t))$ back onto z. We used a simple squared loss to train the style encoder:

$$\mathcal{L}_{style} = \mathbb{E}_{t,z\sim\mathcal{N}(0,1)}||z - S(G(z, \varphi(t)))||_2^2 \qquad (6)$$

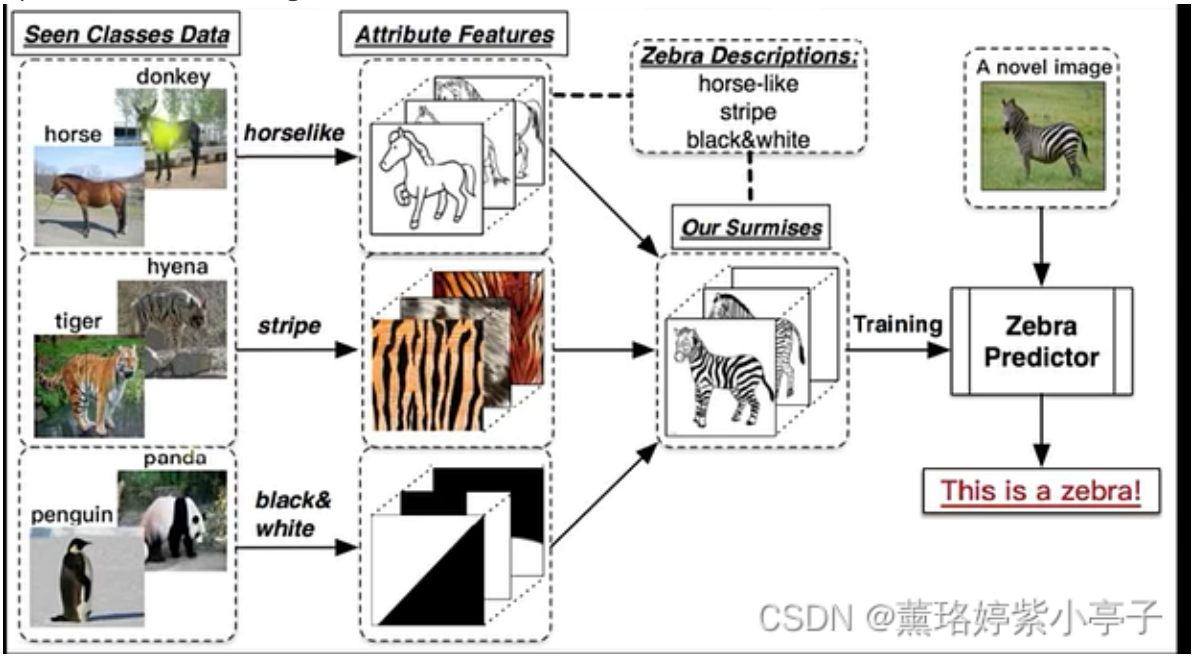
where S is the style encoder network. With a trained generator and style encoder, style transfer from a query image x onto text t proceeds as follows:

$$s \leftarrow S(x), \hat{x} \leftarrow G(s, \varphi(t))$$

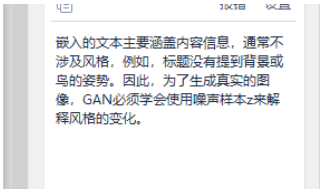
where $\hat{x}$ is the result image and s is the predicted style.

随后翻转G进行风格迁移

tips： zero-shot learning 零样本学习



The text embedding mainly covers content information and typically nothing about style, e.g. captions do not mention the background or the bird pose. Therefore, in order to generate realistic images then GAN must learn to use noise sample z to account for style variations.



为了量化对幼崽的解缠程度，我们设置了两个以噪声z作为输入的预测任务：姿态鉴别和色彩鉴别

2. 学习了sftp和xftp的使用
3. 配置了服务器一个