Spherical CNNS

Liangjie Cao

July 12, 2018

1. Background

One of the most common and challenging problems in Natural Language Processing and Computer Vision is that of image captioning: given an image, a text description of the image must be produced. Text to image synthesis is the reverse problem: given a text description, an image which matches that description must be generated. From a high-level perspective, these problems are not different from language trans- lation problems.

This pink flower has overlapping petals and yellow florets growing in the middle.

This bird is white with blue on its back and has a long, pointy beak.

Figure 1. A sample from the Oxford-102 dataset (left) and CUB-200 dataset (right), together with one of their associated descriptions collected by Reed *et al.* [2]

In the same way similar semantics can be encoded in two different languages, images and text are two different languages to encode related information. Nevertheless, these problems are entirely different because text-image or image-text conversions are highly multimodal problems. If one tries to translate a simple sentence such as "This is a beautiful red flower" to French, then there are not many sentences which could be valid translations. If one tries to produce a mental image of this description, there is a large number of possible images which would match this description. Though this multimodal behaviour is also present in image captioning problems, there the problem is made easier by the fact that language is mostly sequential. This structure is exploited by conditioning the generation of new words on the previous (already generated) words. Because of this, text to image synthesis is a harder problem than image captioning. The generation of images from natural language has many possible applications in the future once the technology is ready for commercial applications.

People could create customised furniture for their home by merely describing it to a computer instead of spending many hours searching for the desired design. Content creators could produce content in tighter collaboration with a machine using natural language.

2. Datasets

The publicly available datasets used in this report are the Oxford-102 flowers dataset [1] and the Caltech CUB-200 birds dataset [3]. These two datasets are the ones which are usually used for research on text to image synthesis. Oxford-102 contains 8,192 images from 102 categories of flowers. The CUB-200 dataset includes 11,788 pictures of 200 types of birds. These datasets include only photos, but no descriptions. Nevertheless, the authors used the publicly available captions collected by Reed *et al.* [4] for these datasets using Amazon Mechanical Turk. Each of the images has five descriptions. They are at least ten words in length, they do not describe the background, and they do not mention the species of the flower or bird (Figure 1). Because the number of images is small, the author perform data augmentation. He apply random cropping and random left-right flipping of the images and split the datasets into train and test datasets such that they contain disjoint classes of images. The datasets are summarised in Table 1.

Dataset/Number of image	Train	Test	Total
Flowers	7034	1155	8192
Augmented flowers (256x256)	675264	110880	786432
Birds	8855	2933	11788
Augmented birds (256x256)	850080	281568	1131648

Table 1. Summary statistics of the datasets

3. Generative Models

The task of text to image synthesis perfectly fits the description of the problem generative models attempt to solve. The current best text to image results are obtained by Generative Adversarial Networks (GANs), a particular type of generative model. Before introducing GANs, generative models are briefly explained in the next few paragraphs.

Before defining them, there are some necessary notation. Consider a dataset $X = x^{(1)}, ..., x^{(m)}$ composed of m samples where $x^{(i)}$ is a vector. In the particular case of this report, $x^{(i)}$ is an image encoded as a vector of pixel values. The dataset is produced by sampling the images from an unknown data generating distribution $\mathbb{P}_r$, where r stands for real. One could think of the data generating distribution as the hidden distribution of the Universe which describes a particular phenomenon. A generative model is a model which learns to generate samples from a distribution $\mathbb{P}_g$ which estimates $\mathbb{P}_r$. The model distribution, $\mathbb{P}_g$, is a hypothesis about the true data distribution $\mathbb{P}_r$.

Most generative models explicitly learn a distribution $\mathbb{P}_g$ by maximising the expected log-likelihood $\mathbb{E}_{X\mathbb{P}_r} log(\mathbb{P}_g(x \parallel \theta))$ with respect to θ, the parameters of the model. Intu-itively, maximum likelihood learning is equivalent to putting more probability mass around the regions of X with more examples from X and less around the regions with fewer examples. It can be shown that the log-likelihood maximisation is equivalent to minimising R the Kullback-Leibler divergence $KL(\mathbb{P}_r \parallel \mathbb{P}_g) = \int_X \mathbb{P}_r log \frac{\mathbb{P}_r}{\mathbb{P}_g} dx$ assuming $\mathbb{P}_r$ and $\mathbb{P}_g$ are densities. One of the valuable properties of this approach is that no knowledge of the unknown $\mathbb{P}_r$ is needed because the expectation can be approximated with enough samples according to the weak law of large numbers.

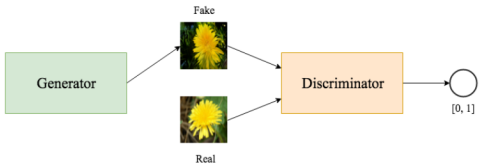

Figure 2. High-level view of the GAN framework. The generator produces synthetic images. The discriminator takes images as input and outputs the probability it assigns to the image of being real. A common analogy is that of an art forger (the generator) which tries to forge paintings and an art investigator (the discriminator) which tries to detect imitations

4. GAN

The GAN framework is based on a game played by two entities: the discriminator (also called the critic) and the generator. Informally, the game can be described as follows. The generator produces images and tries to trick the discriminator that the generated images are real. The discriminator, given an image, seeks to determine if the image is real or synthetic. The intuition is that by continuously playing this game, both players will get better which means that the generator will learn to generate realistic images (Figure 2).

References

[1] M. E. Nilsback and A. Zisserman. Automated flower classification over a large number classes. In *ICCV*, 2009. 1

[2] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 1

[3] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 1

[4] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds200-2011 dataset. *California Institute of Technology*, 2011. 1