Controllable content generator

Cédric Beaulac, David K. Duvenaud & Jeffrey S. Rosenthal

Department of Statistical Sciences, University of Toronto

Objectives

We want to use the recent advances in generative models to construct *controllable* content generators that produce *valuable* content. We would like to :

- Establish the specific needs of the field.
- Define controllability.
- Review and update current techniques.

This would democratize artistic content creation.

Introduction

The algorithmic creation of content allows to generate an infinite amount of diverse content in an inexpensive way. It has also great storage benefit.

Procedural content generation algorithms are part of almost every video games and movies. The content generated in a video game can take **multiple forms**: story elements, non-playable characters, art assets and more. For the purpose of this poster, we will focus on art assets such as, side walks, buildings, trees and skies.

Can we create content generators useful to artists, game designers and other content creators ? We put our effort on giving *control* to the content creator so that they can truly create what they envision.

Let us focus on the simple task of generating pictures of skies:

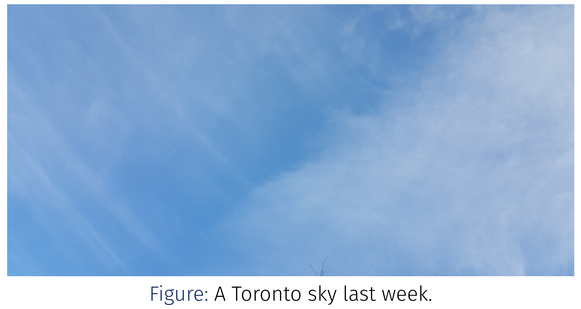


Figure: A Toronto sky last week.

Control

We are attempting to rigorously define *control*. Intuitively, we define *control* along four axis :

- **Size** : How many properties ?
- **Strength** : How much the output is affected ?
- **Range** : Can we inter/extrapolate ?
- **Disentanglement/Interpretability**

We are currently working on establishing metrics to evaluate the controllability of generators. They could be used to compare algorithms or as part of the optimization problem.

For example, the **mutual information** between the control variables and the generated image could be used as a metric to evaluate the control strength.

Generative models

Generative Adverserial Networks (GAN) and Variational AutoEncoders (VAE)are the state-of-the-art machine learning techniques to generate images. They share a similar architecture:

- They draw samples from a lower dimension latent space.
- They process latent space samples through a neural network (NN) to generate images.

Why are these models using a lower-dimension latent space ?

It makes the generative process extremely fast; It is much faster to generate a sample from a low-dimensional space and then apply a function to this sample, than generating an image containing a large amount of correlated pixels.

We assume that the variability in the images can be explained with a small set of generative factors and that everything else is noise. For our example : the type of clouds, the time of the day and the cloud density could be considered our generative factors.

Autoencoder

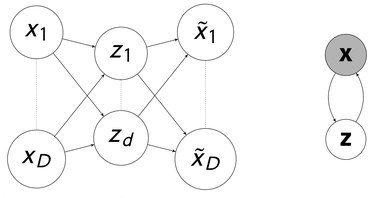
An autoencoder learns not only how to generate images from low-dimension representations, but also attempts to learn these representations from the data. More precisely, assuming we have :

- **x** : An observation (image) of size D.
- **z** : A latent representation (code) of size $d << D$.

An autoencoder aims to learn an encoding function $q : \mathcal{X} \to \mathcal{Z}$ and a decoding function $p : \mathcal{Z} \to \mathcal{X}$ simultaneously.

These functions can take multiple forms and we can define various optimization objective functions.

Example: Functions are linear and we minimize mean square reconstruction error.



The solution to this system is PCA. We can learn more complex encoding and decoding functions using NNs.

Variational autoencoder

Assuming a distribution for **x** and **z** leads to a straight forward sampling technique.

The model :

- A prior on $p_\theta(\mathbf{z})$ (Usually isotropic Normal)
- A decoding distribution $p_\theta(\mathbf{x}|\mathbf{z})$ ($\theta = NN_1(\mathbf{z})$)
- An encoding distribution $q_\varphi(\mathbf{z}|\mathbf{x})$ ($\varphi = NN_2(\mathbf{x})$) that serves as an approximation for the posterior $p_\theta(\mathbf{z}|\mathbf{x})$

The system is optimized using maximum likelihood. More precisely, we maximize the Evidence Lower BOunds (ELBO), which is a lower bound of the observed data log-likelihood :

$$\log p_\theta(x) \geq \mathbf{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\varphi(\mathbf{z}|\mathbf{x})] \quad (1)$$

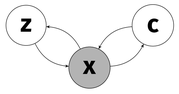
Our strategy

Our strategy is to use the latent space to control the produced image by sampling from a subspace, using an alternative sampling distribution or modifying the objective function accordingly. Our proposed solutions are inspired by :

- Semi-supervised learning,
- Disentangled representation, and
- Importance sampling.

Work in progress

We are currently implementing a model inspired by semi-supervised learning. This model has two latent spaces :



where **c** is a set of control variables. They are generative factors the content creators want to control. For example, the user would identify the cloud type and the cloud density of a few pictures of skies for the learning procedure.

The generating procedure is quite fast. The user selects the value for all control variables **c**, samples from **z** and then uses these variables to generate the image.

References

[1] Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. *CoRR*, abs/1802.04942, 2018.

[2] Diederik P. Kingma. *Variational Inference & Deep Learning : A New Synthesis*. PhD thesis, Universiteit van Armsterdam, 10 2017.

[3] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.