EmotiGAN: Text-To-Emojis With Generative Adversarial Networks

Marcel Puyat

Introduction

Emojis have become ubiquitous in modern means of communication. In this project, we take on the task of generating an emoji given textual input using a Conditional Generative Adversarial Network[1] (GAN). Training a Conditional GAN to come up with varying outputs given the same input is an unsolved problem. We use a pre-trained *word2vec*[2] model, a network architecture based on Radford et. al's Deep Convolutional GAN[3], and come up with a novel training mechanism that results in EmotiGAN: a model that converts words to emojis and is able to generate varying kinds of styles even when given the same textual inputs.

Objective Function

To stabilize training, we add decaying instance noise[5] to images in order to avoid undefined KL-divergence between fake and real data at the start of training. Discriminator objective:

$$log(D(\mathbf{x}+\varepsilon)) + log(1 - D(G(\mathbf{z}+\varepsilon)))$$

We use Goodfellow's non-saturating generator objective function[4]:

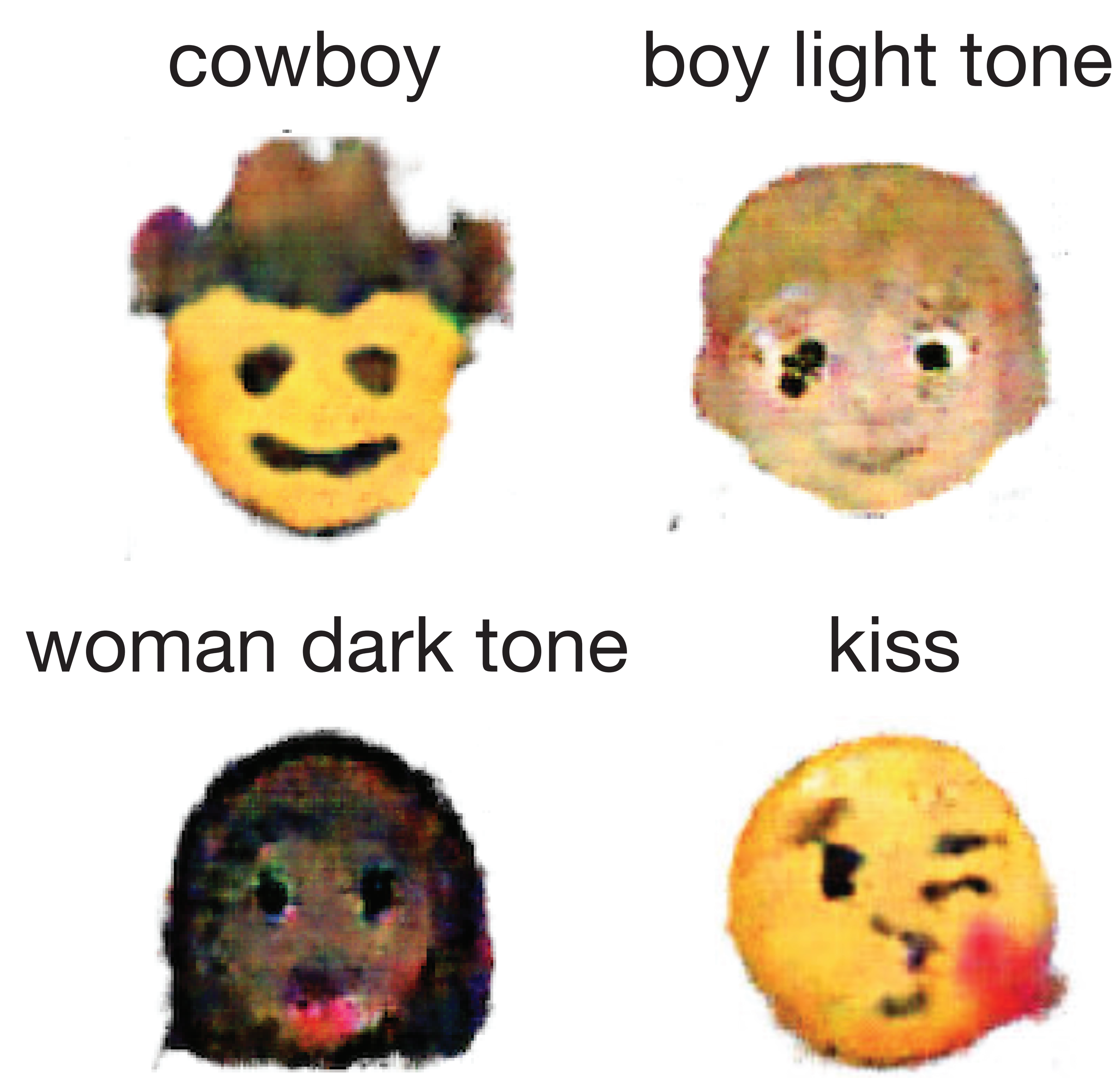
$$-log(D(G(\mathbf{z}))$$

References

[1] M. Mirza et al. Conditional generative adversarial nets.
[2] T. Mikolov et al. Efficient Estimation of Word Representations in Vector Space.
[3] Radford et al. Unsupervised representation learning with deep convolutional generative adversarial networks.
[4] Goodfellow et al. Improved techniques for training gans.
[5] Caballero et al. Amortised MAP Inference for Image Super-resolution.

Dataset & Results

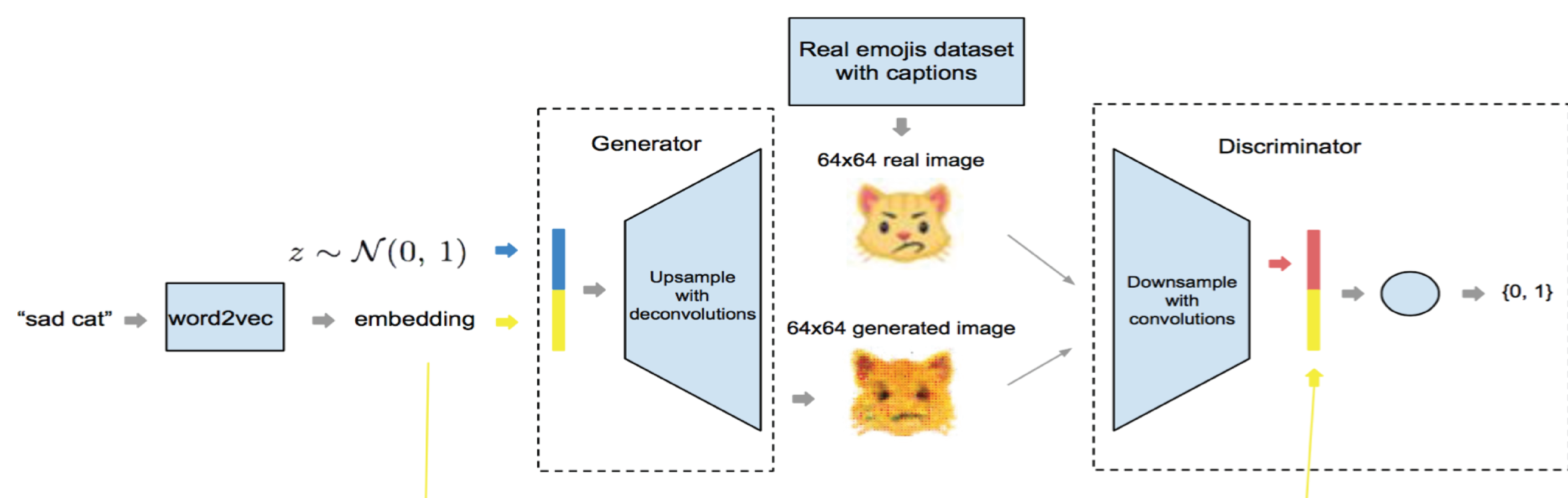
We trained on a dataset of ~5000 emojis from unicode.org, where each emoji had a label describing it. After training for ~15k iterations (Tensorflow on Paperspace Cloud GPU, NVIDIA Quadro P6000), we came up with plausible emojis that match the text labels provided as input. Qualitative evaluation is one of the main metrics with unsupervised image generation methods. Nonetheless, we also used the inception score[4] metric to compare vanilla DCGAN to EmotiGAN with its novel training method and saw a significant delta of around 2x.



Training Method

Our solution to coming up with varying styles is simple yet elegant: a training algorithm that trains a discriminator using minibatch-discrimination[4] over batches with the same input but different-looking emoji styles.

Network Architecture



Future Work

- Our training method to generate varying outputs relies on batches with the same input, which doesn't scale well as training size increases. A clear direction is to try batches with N repeated inputs within the batch.
- Varying Conditional GAN outputs isn't specific to emojis. This should be tested on other datasets.