Image Generation with a Variational Autoencoder

Marissa Douglas

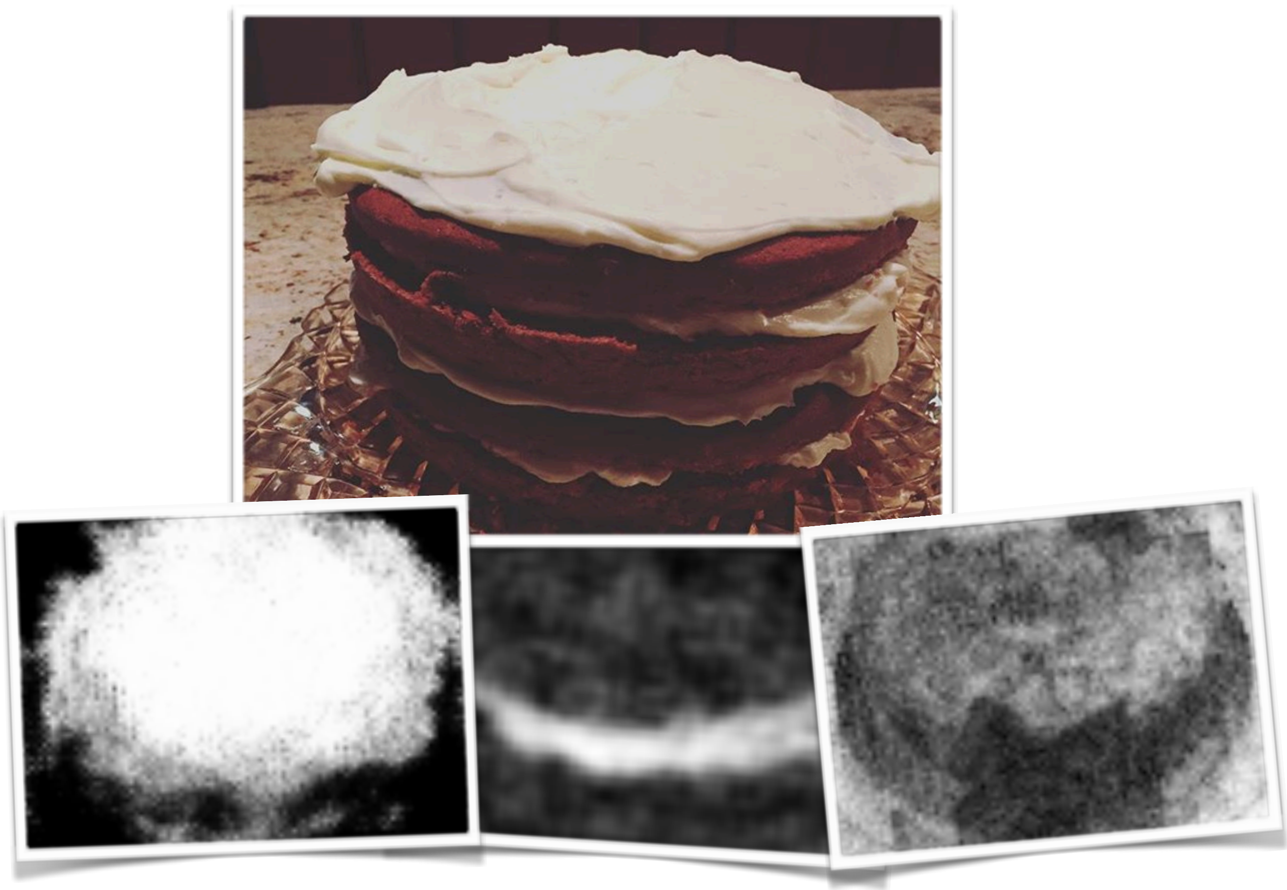


Image Generation with a Variational Autoencoder

Abstract

Images are synthesized via a convolutional auto-encoding neural network where randomness is inserted between the encoding and decoding portions of the network to reconstruct images which deviate from the input set.

Introduction

Image generation has many useful applications primarily in the realm of product development. Although auto-encoding neural networks are not the ideal tool for new image generation due to information loss in the encoding process, the structure of such neural networks is easily understood and the quality of the output can easily be measured via a visual comparison of the input and output images. Therefore, this project utilizes a convolutional variational autoencoder to synthesize images of cake.

Data Acquisition

Images URLs were downloaded from ImageNet [2], specifically the 'Ice -cream cake, icebox cake' synset. Many of these URLs were no longer live, necessitating a manual screening step after URLs were scraped. Empty files were deleted from the set.

Additional images and image URLs were taken from thepictaram.club and Google Images. The web-sourced image set was subjected to a final manual sorting step to include only round, single-tiered cakes in the final set to be used in the model. There were 431 images in this set.

Preprocessing

Images were converted to grayscale and sized to 100x100 pixels by the following process:

1. For images with unequal height and width dimensions, the larger dimension was cropped to match the smaller dimension. Although not a robust operation, the result of this operation did not affect any image's suitability for inclusion in the dataset.

2. Images with equal height and width (original or cropped) were resized to 100x100 pixels by OpenCV INTER_LINEAR interpolation method.

Pixels were standardized to have values between 0 and 1.

Model Design and Training

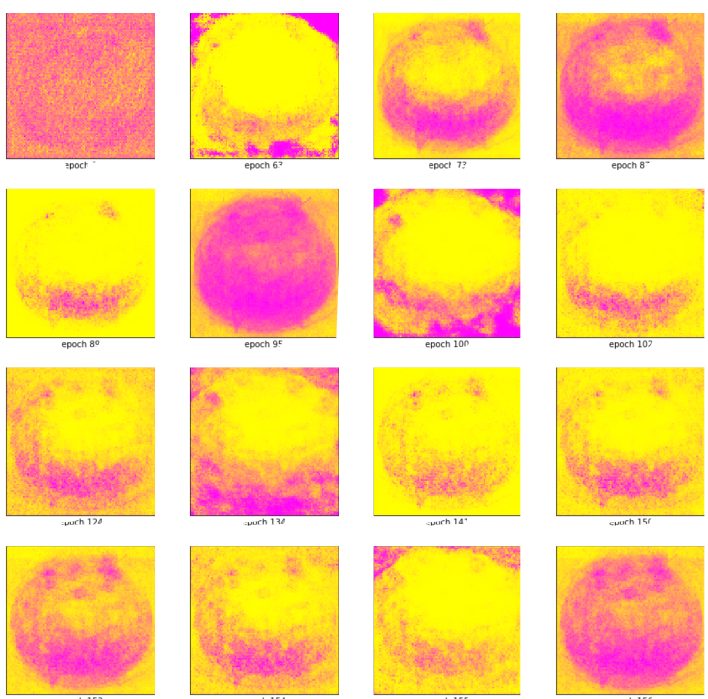
The model was adapted from code from the Keras blog [1], which was built with the Keras library for the MNIST dataset.

Adaptations to the model for the assembled cake dataset:

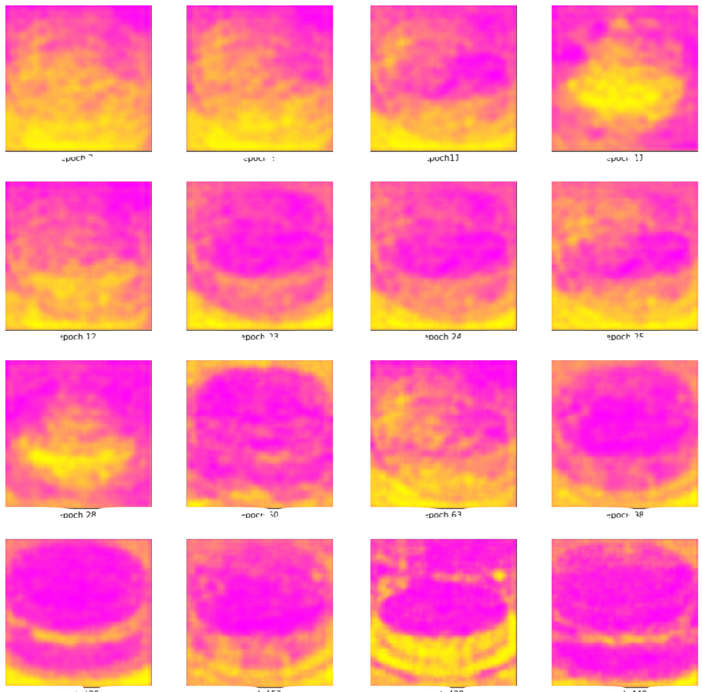
- The number of dimensions to reduce the inputs to was increased to 150.
- The loss function from F. Chollet's blog post [1] originally includes a Kullback-Leibler regularization term, but this was not used in the trials of this effort (see "Future Work"). Instead, the model used simple binary cross entropy.
- In lieu of the RMSprop, Adam was used as the optimizer with a learning rate of 0.0001. The default learning rate (0.001) caused the model to diverge.
- The final iteration of the model was set to stop when the validation data loss stopped reducing for 25 epochs. This model trained for 156 epochs before automatically stopping due to this patience parameter.
- The model returned a randomly generated image on every epoch and a reconstructed input image every 10 epochs for benchmarking purposes.

Generated Images

(1) The images below are representative outputs over the course of the final model fitting.



(2) The images below are representative outputs when the model was allowed to train with all images without validation data. The loss remained steadily decreasing after 500+ epochs.



Conclusions

When the model is allowed to overfit, as in the second set of images above, the output images are more recognizable as cakes but are close matches to the input images. In the case where the model stops when the validation loss is minimized (first set of images above), the output is more abstract but still cake-like. Greater authenticity in the validation loss-minimized set may be achieved with a larger data set. Old links from the ImageNet site and barriers to web scraping put in place by image search engines prevented further augmentation of the original dataset in an efficient manner.

For the non-overfit images, it is obvious the 'learning' operation is still underway and would benefit from more input images to "smooth-out" the distribution from which the generator model can sample from.

Future Work

(1) **Increase input data**. As mentioned, above the generator model would benefit from a larger dataset. New image sources may provide more images of the same subject user here or the model could also be trained to generate images of a subject for which there is a larger synset of images available from ImageNet.

(2) **Regularization.** The Kullback-Leibler regularization term, which was not used in the loss function, may be added back into the custom loss function presented in the Keras blog [1]. The addition of the regularization term may afford the model better learning power while preventing overfitting.

References

[1] Chollet, Francois. “Building Autoencoders in Keras.” *The Keras Blog*, 14 May 2016, blog.keras.io/building-autoencoders-in-keras.html

[2] “ImageNet Download.” *ImageNet*, Stanford Vision Lab, Stanford University, Princeton University; 2016, image-net.org/download