Machine Learning Engineer Nanodegree

Capstone Proposal

Surya Krishnamurthy
August 5th, 2018

Neural Style Transfer

Domain Background

Computer vision is a field that has been gaining a lot of momentum in recent times. It powers many emerging technologies like facial recognition, self-driving cars and more. With the numerous breakthroughs in the field of AI and the increase in the performance of computers today, applying computer vision to industrial applications is easier than ever. The invention of the Convolutional Neural Networks (CNN), especially, lead to several possibilities.

CNNs possess interesting properties that arise out of its ability to learn higher level features from an image. Competitions like ImageNet have given the community several architectures that surpass human level performance on tasks like image classification. This further allows easy research and development through transfer learning from models trained on these competition datasets.

For this project, we particularly concentrate on style transfer. In style transfer, we render a content image in different styles. Several studies exploring how to automatically turn images into synthetic artworks have been performed since the mid-1990s. But, impressive results have been produced only after the post-neural era.

Problem Statement

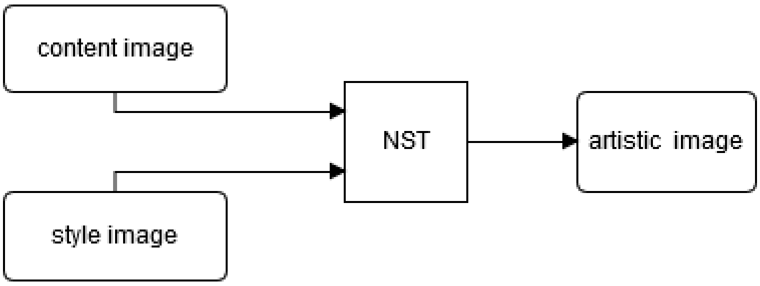
In this project, we will develop an artificial system based on a Deep Neural Network that creates artistic images of high perceptual quality. The algorithm will also help us develop an understanding of how humans create and perceive artistic imagery.
Given an image, the algorithm will separate and recombine the content from the image and style of an arbitrary image using neural representations.
The algorithm will be evaluated based on some quantitative metrics and qualitatively by visualizing the images generated by the algorithm.

Datasets and Inputs

- **Dataset.** The algorithm depends on the VGG architecture pre-trained on a complex classification task (in our case the ImageNet dataset). So, no datasets were used to train the architecture. The algorithm is also a one-shot generation process - it doesn't require iteration through several images.
- **Inputs.** The algorithm requires 2 major inputs - A content image and a style image whose style will be applied to the content image.

Solution Statement

We use **Neural Style transfer** for the problem. In Neural style transfer, we take a style image and apply its style, texture, and other artistic patterns, to a content image to generate a new visually pleasing image. Unlike conventional image filters that transform the image in the color space, style transfer alters the entire style of the content image while preserving the semantic content.



The method was first proposed by Gatys *et al* (2015). They proposed to model the content of a photo as the feature responses from a pre-trained CNN, and further model the style of an artwork as the summary feature statistics. The key idea behind their algorithm is to iteratively optimise an image with the objective of matching desired CNN feature distribution, which involves both the photo's content information and artwork's style information.

Benchmark Model

- **Random model.** We expect the model to perform better in creating styled images than a model that generates a random image or applies some random changes to a given content image. We expect to easily accomplish this task.
- **The original paper.** Our main benchmark is to generate images that offer visual experiences similar to that in the original paper shown below. But, our choice of inputs is different from the original paper.

Evaluation Metrics

This paper (section 6.3) built a standardized benchmark to quantitatively evaluate this class of algorithms. The primary evaluation metrics are as follows:

1. Training time
2. Loss comparison with iteration
    a. Total loss
    b. Content loss
    c. Style loss

Further, we qualitatively evaluate the generated image saved during the checkpoints in our model to ensure that the algorithm is indeed producing the expected results. This evaluation relies on the aesthetic judgment of observers and is the criteria for stopping metric.

Project Design

Programming language

Python 3.6

Programing libraries

- Keras
- Scipy
- Numpy
- Pillow
- Matplotlib

Machine Learning Algorithm design

CNN Architecture:
We will use the pretrained VGG network from keras

```
keras.applications.vgg16.VGG16(include_top=True, weights='imagenet')
```

Loss function:
In the algorithm, we ultimately minimize the distance of the canvas image from the content representations of the photograph in one layer of the network and the style representations of the painting in a number of layers of the CNN. The loss functions are defined below and will be implemented using the keras deep learning framework.

Content loss:

$$\mathcal{L}_{content}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l} - P_{ij}^{l}\right)^2 .$$

Style loss:
For the style loss we compute the correlation between the different filter responses. The feature correlations are given by the Gram matrix.

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}.$$

And the contribution of each style representation layer to total loss is

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}\left(G_{ij}^{l} - A_{ij}^{l}\right)^2$$

$$\mathcal{L}_{style}(\vec{a},\vec{x}) = \sum_{l=0}^{L} w_l E_l$$

Total loss:

$$\mathcal{L}_{total}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{content}(\vec{p},\vec{x}) + \beta\mathcal{L}_{style}(\vec{a},\vec{x})$$

Optimizer:
As opposed to the trivial SGD algorithm, we will be using the **Limited-memory Broyden–Fletcher–Goldfarb–Shanno** algorithm as it is shown to converge faster for this

particular type of problem. We will use the scipy optimizer module to use the L-BFGS algorithm.

```
scipy.optimize.fmin_l_bfgs_b(loss_func, x0, fprime)
```

Evaluation:
A checkpointing scheme will be implemented which will save the generated image and loss once every few epochs which can be used for evaluation purposes.

References

Gatys, L.A., Ecker, A.S., & Bethge, M. (2015). A Neural Algorithm of Artistic Style. CoRR, abs/1508.06576.
Jing, Y., Yang, Y., Feng, Z., Ye, J., & Song, M. (2017). Neural Style Transfer: A Review. CoRR, abs/1705.04058.
https://towardsdatascience.com/a-brief-introduction-to-neural-style-transfer-d05d0403901d
https://blog.slavv.com/picking-an-optimizer-for-style-transfer-86e7b8cba84b
https://medium.com/mlreview/making-ai-art-with-style-transfer-using-keras-8bb5fa44b216
https://www.bonaccorso.eu/2016/11/13/neural-artistic-style-transfer-experiments-with-keras