**Deep Learning in Computer Vision and Visual Arts**

**Learning Artistic Style with the Gram Matrix of Feature Maps in a Convolutional Neural Network**

Philippe Paradis; Yue Dong

Department of Mathematics and Statistics, University of Ottawa

---

**Introduction**

Gatys et al. [1] recently demonstrated that deep convolutional neural networks could generate artistic stylized images from a single artist's painting example and an arbitrary image. This project conducts an evaluation of this convolutional neural networks (CNN)-based characterization of the concept of texture or style and attempts to understand how the Gram matrices of feature maps capture those concepts so accurately. We also used a faster and less memory demanding model called Network In Network (NIN), instead of VGG-19 as in the original paper.

**Convolutional neural networks**

CNN consist of alternating convolutional layers and pooling layers. Convolution layers take inner product of linear filters followed by a nonlinear activation function to scan the input. The resulting outputs are called feature maps.

**NIN is obtained by replacing the linear convolutional filter in CNN with a more potent nonlinear function approximator - the micro neural network with a multilayer perceptron.**

**Methodology I: Content Representation**

- Preserve objects and their arrangements (content) by constructing a loss function that attempts to match feature maps directly.

Let \( p \) and \( b \) be the original image and the image that is generated and \( P \) and \( F \) their respective feature representation in layer \( \ell \). We then define the squared-error loss between the two feature representations

\[
L_{\text{content}}(p, F, \ell) = \frac{1}{2} \sum_{r,s} (F_{r,s} - P_{r,s})^2
\]

The derivative of this loss with respect to the activations in layer \( \ell \) equals

\[
\frac{\partial L_{\text{content}}}{\partial F} = \begin{cases} (F_r - P_r) & \text{if } F_{r,s} > 0 \\ 0 & \text{if } F_{r,s} < 0 \end{cases}
\]

from which the gradient with respect to the input \( x \) can be computed using standard error back-propagation.

**Methodology II: Style Representation**

- Instead of preserving feature maps directly, preserve style through *Gram matrices*, which is defined as the dot products of all feature maps (at each layer)
- The spatial information is lost when taking dot products
- The dot products preserve filters with strong activations, hence capture important elements but not their locations

Let \( a \) and \( b \) be the original image and the image that is generated and \( A \) and \( G \) their respective style representations (Gram matrices) in layer \( \ell \). We then define the mean squared-error loss between the two

\[
E_{\ell} = \frac{1}{N \times M} \sum (A_{r,s} - G_{r,s})^2
\]

and the total loss is the weighted sum of losses from multiple layers

\[
L_{\text{style}}(a, b) = \sum_{\ell} \beta E_{\ell}
\]

The derivative of \( E_{\ell} \) with respect to the activations in layer \( \ell \) equals

\[
\frac{\partial E_{\ell}}{\partial A} = \frac{1}{2} \sum (P_{r,s} (C - A))_{r,s} \quad \text{if } F_{r,s} > 0
\]

and

\[
\frac{\partial E_{\ell}}{\partial A} = 0 \quad \text{if } F_{r,s} < 0
\]

The gradients of \( E_{\ell} \) with respect to the activations in lower layers of the network can be readily computed using standard error back-propagation.

**Methodology III: Combine Content and Style**

The loss function we minimize is a linear combination of the content loss function and the style loss function:

\[
L_{\text{total}}(\tilde{p}, a, b) = \alpha L_{\text{content}}(\tilde{p}, b) + \beta L_{\text{style}}(a, \tilde{b})
\]

**Results & Conclusions**

In a way, the style and content of an image can be distinguished by a deep CNN. This project proposes a faster implementation with NIN to draw various images with artistic styles. The resulting network could produce comparable results to Gatys et al., albeit of slightly lower quality, but requires less GPU memory by about a factor of 10 times (making it possible to use it on commodity low-end GPU hardware; in our case we used an nVidia GTX 570 with 1GB of memory).

---

**Acknowledgments**

We would like to thank many wonderful individuals including Dr. Nathalie Japkowicz and Dr. Vladimir Pestov for their invaluable contributions and supports.

---

**Bibliography**


**Further Information**

Yue Dong: M.Sc. Candidate, Department of Mathematics and Statistics, University of Ottawa, ydong029@uottawa.ca

Philippe Paradis: philippe.paradis@gmail.com