

Demystifying Neural Style Transfer

一、创新点

Neural Style Transfer (NST , 神经风格迁移) 已经被研究得非常多了，但是仍然存在一些没有被说明白的问题，其中最重要的一个问题就是：为什么Gram矩阵能够表示风格？这篇文章就对这个问题提出了一个全新的解释。

作者把NST问题当做一个**domain adaptation**的问题。具体来说，作者发现Gram矩阵的匹配问题实际上等同于最小化二阶多项式核的Maximum Mean Discrepancy (MMD) ，而MMD是一种分布差异性的度量，因此NST的本质就是匹配风格图像和生成图像的特征分布。

二、基本概念解释

1. Domain Adaptation

Domain adaptation属于迁移学习的一种。它的目标就是将在source domain学习到的模型迁移到target domain上，而且target domain通常是没有label的。domain adaptaiton很重要的一点就是要使source domain和target domain分布的差异性最小。那么问题来了，这个分布的差异性要怎么衡量？一种很常用的差异性度量方法就是Maximum Mean Discrepancy (MMD , 最大平均差异) , 它在Reproducing Kernel Hilbert空间中衡量了样本均值的差异性。

参考：https://www.sohu.com/a/227995138_642762

2. Maximum Mean Discrepancy

假设我们有两个样本集，分别表示为 $X = \{x_i\}_{i=1}^n, Y = \{y_j\}_{j=1}^m$ ，其中 x_i 和 y_j 分别表示从分布 p 和 q 产生的数据。

那么MMD则定义为：

$$\begin{aligned} & \text{MMD}^2[X, Y] \\ &= \|\mathbf{E}_x[\phi(\mathbf{x})] - \mathbf{E}_y[\phi(\mathbf{y})]\|^2 \\ &= \left\| \frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i) - \frac{1}{m} \sum_{j=1}^m \phi(\mathbf{y}_j) \right\|^2 \\ &= \frac{1}{n^2} \sum_{i=1}^n \sum_{i'=1}^n \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_{i'}) + \frac{1}{m^2} \sum_{j=1}^m \sum_{j'=1}^m \phi(\mathbf{y}_j)^T \phi(\mathbf{y}_{j'}) \\ &\quad - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m \phi(\mathbf{x}_i)^T \phi(\mathbf{y}_j), \end{aligned}$$

其中 $\phi(\cdot)$ 是一种特征映射函数。如果使用kernel function : $k(x, y) = \langle \varphi(x), \varphi(y) \rangle$, 则上式可以表示为 :

$$\begin{aligned}
& \mathbf{MMD}^2[X, Y] \\
= & \frac{1}{n^2} \sum_{i=1}^n \sum_{i'=1}^n k(\mathbf{x}_i, \mathbf{x}_{i'}) + \frac{1}{m^2} \sum_{j=1}^m \sum_{j'=1}^m k(\mathbf{y}_j, \mathbf{y}_{j'}) \\
& - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(\mathbf{x}_i, \mathbf{y}_j).
\end{aligned}$$

kernel function隐式地将样本从一个低维特征空间映射到一个更高维的特征空间。

三、重新理解NST

1. 回顾NST

NST要做的事情就是，给你一个内容图片 x_c 和一个风格 x_s 图片，你去产生一个新的图片 x^* 。它们对应的feature maps的矩阵表示分别为： $P^l \in R^{N_l \times M_l}$, $S^l \in R^{N_l \times M_l}$, $F^l \in R^{N_l \times M_l}$, N_l 表示 l 层feature maps。

$M_l = feature_map_width * feature_map_height$.

NST就是通过优化以下目标函数来生成合成图像 x^* :

$$\mathcal{L} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style},$$

其中 α 和 β 分别表示内容和风格loss的权重。

其中，内容损失为：

$$\mathcal{L}_{content} = \frac{1}{2} \sum_{i=1}^{N_l} \sum_{j=1}^{M_l} (F_{ij}^l - P_{ij}^l)^2,$$

风格损失为：

$$\mathcal{L}_{style} = \sum_l w_l \mathcal{L}_{style}^l,$$

其中， w_l 表示为各层loss的权重。注意，风格损失包括多层，每一层的损失为：

$$\mathcal{L}_{style}^l = \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} (G_{ij}^l - A_{ij}^l)^2,$$

其中合成图片的Gram矩阵：

$$G_{ij}^l = \sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l,$$

A_{ij}^l 为风格图片对应的Gram矩阵，定义和Gram矩阵类似。

2. 重新形式化风格loss

风格loss：

$$\mathcal{L}_{style}^l = \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} (G_{ij}^l - A_{ij}^l)^2,$$

可以进行转化：

$$\begin{aligned} \mathcal{L}_{style}^l &= \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \left(\sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l - \sum_{k=1}^{M_l} S_{ik}^l S_{jk}^l \right)^2 \\ &= \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \left(\left(\sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l \right)^2 + \left(\sum_{k=1}^{M_l} S_{ik}^l S_{jk}^l \right)^2 - 2 \left(\sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l \right) \left(\sum_{k=1}^{M_l} S_{ik}^l S_{jk}^l \right) \right) \\ &= \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} (F_{ik_1}^l F_{jk_1}^l F_{ik_2}^l F_{jk_2}^l + S_{ik_1}^l S_{jk_1}^l S_{ik_2}^l S_{jk_2}^l - 2 F_{ik_1}^l F_{jk_1}^l S_{ik_2}^l S_{jk_2}^l) \\ &= \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} (F_{ik_1}^l F_{jk_1}^l F_{ik_2}^l F_{jk_2}^l + S_{ik_1}^l S_{jk_1}^l S_{ik_2}^l S_{jk_2}^l - 2 F_{ik_1}^l F_{jk_1}^l S_{ik_2}^l S_{jk_2}^l) \\ &= \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \left(\left(\sum_{i=1}^{N_l} F_{ik_1}^l F_{ik_2}^l \right)^2 + \left(\sum_{i=1}^{N_l} S_{ik_1}^l S_{ik_2}^l \right)^2 - 2 \left(\sum_{i=1}^{N_l} F_{ik_1}^l S_{ik_2}^l \right)^2 \right) \\ &= \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \left((\mathbf{f}_{\cdot k_1}^l)^T \mathbf{f}_{\cdot k_2}^l + (\mathbf{s}_{\cdot k_1}^l)^T \mathbf{s}_{\cdot k_2}^l - 2 (\mathbf{f}_{\cdot k_1}^l)^T \mathbf{s}_{\cdot k_2}^l \right), \end{aligned}$$

其中 $\mathbf{f}_{\cdot k}^l$ 和 $\mathbf{s}_{\cdot k}^l$ 分别表示 \mathbf{F}^l 和 \mathbf{S}^l 的第 k 列。如果使用二阶多项式核： $k(x, y) = (x^T y)^2$ ，则上式可以简单地表示为：

$$\begin{aligned}
\mathcal{L}_{style}^l &= \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \left(k(\mathbf{f}_{\cdot k_1}^l, \mathbf{f}_{\cdot k_2}^l) \right. \\
&\quad \left. + k(\mathbf{s}_{\cdot k_1}^l, \mathbf{s}_{\cdot k_2}^l) - 2k(\mathbf{f}_{\cdot k_1}^l, \mathbf{s}_{\cdot k_2}^l) \right) \\
&= \frac{1}{4N_l^2} \text{MMD}^2[\mathcal{F}^l, \mathcal{S}^l],
\end{aligned}$$

其中

$$\mathcal{F}^l$$

和

$$\mathcal{S}^l$$

的分别代表 \mathcal{F}^l 和 \mathcal{S}^l 的列构成的集合。这样，实际上把feature maps中每一个位置对应的向量当成了一个样本。因此，style loss忽略了特征的位置。

以上的式子表明：

1. 图像的风格本质上可以表示成CNN层的特征分布。
2. 风格迁移可以看成是分布对齐的过程。

The style transfer can be seen as a distribution alignment process from the content image to the style image

3. 使用不同的adaptation方法做NST

以上的解释表明NST被当做分布对齐问题，这个在domain adaptation中也是很重要的一个问题。如果我们把一张图像在CNN中某一层的风格当做一个domain，风格迁移也能够被当做一个domain adaptation的问题。不过问题的特殊性在于，我们把feature maps上每一个点对应的向量当做了数据样本。

由以上讨论已经知道，匹配Gram矩阵实际上可以被当做一个二阶多项式核的MMD过程。如果使用MMD统计测量风格的差异性，style loss可以表示为：

$$\begin{aligned}\mathcal{L}_{style}^l &= \frac{1}{Z_k^l} \text{MMD}^2[\mathcal{F}^l, \mathcal{S}^l], \\ &= \frac{1}{Z_k^l} \sum_{i=1}^{M_l} \sum_{j=1}^{M_l} \left(k(\mathbf{f}_{\cdot i}^l, \mathbf{f}_{\cdot j}^l) + k(\mathbf{s}_{\cdot i}^l, \mathbf{s}_{\cdot j}^l) - 2k(\mathbf{f}_{\cdot i}^l, \mathbf{s}_{\cdot j}^l) \right),\end{aligned}$$

其中 Z_k^l 是一个归一化项。理论上，不同的 kernel function 隐式地将特征映射到更高维的特征空间，因此，如果使用其他的 kernel function 也可以捕捉不同种类的风格。本篇文章主要采取了三种比较流行的 kernel function。

- (1) Linear kernel: $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$;
- (2) Polynomial kernel: $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + c)^d$;
- (3) Gaussian kernel: $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma^2}\right)$.

对于多项式核，我们只采用 $d=2$ 。注意，匹配 Gram 矩阵就等同于采用了 $c=0$, $d=2$ 的多项式核。

对于高斯核，采用 MMD 的无偏估计，它采样了 M_l 对样本，因此可以达到线性计算复杂度。

BN Statistics Matching

Revisiting batch normalization for practical domain adaptation 这篇文章中讲到，BN 的统计量（均值和方差）能够表示 domain 的显著特征，不同 domain，这些统计量的差别会很大，因此用 BN 统计量来表示不同的 domain。这种表示可以很好地将两个 domain distribution 进行很好地对齐。本篇文章受这个启发，加上前文已经说明风格迁移实际上就是一个 domain adaptation 的问题，作者认为将层的 BN 统计量来表示表示风格也是完全可以的。

因此，可以通过 BN 统计量来构建两个 feature maps 的 style loss：

$$\mathcal{L}_{style}^l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left((\mu_{F^l}^i - \mu_{S^l}^i)^2 + (\sigma_{F^l}^i - \sigma_{S^l}^i)^2 \right),$$

其中， $\mu_{F^l}^i$ 和 $\sigma_{F^l}^i$ 分别表示生成图像 x^* 的第 l 层，第 i 个通道对应的 feature map 的均值和方差。

$$\mu_{F^l}^i = \frac{1}{M_l} \sum_{j=1}^{M_l} F_{ij}^l, \quad \sigma_{F^l}^i = \sqrt{\frac{1}{M_l} \sum_{j=1}^{M_l} (F_{ij}^l - \mu_{F^l}^i)^2},$$

$\mu_{S^l}^i$ 和 $\sigma_{S^l}^i$ 表示的是风格图像，意思同上。

四、实验结果

实验结果主要包括：

1. 实现细节。
2. 将多种风格迁移方法进行融合。

1. 实现细节

- 使用VGG19网络。
- content loss : *relu4_2*
- style loss: *relu1_1, relu2_1, relu3_1, relu4_1, relu5_1*
- w_i : 都设置为1。
- x^* 进行随机初始化。
- 两次迭代改变小于0.5%，则停止迭代。最多迭代1000次。

2. 不同的风格表示

对风格进行特征重构分析。

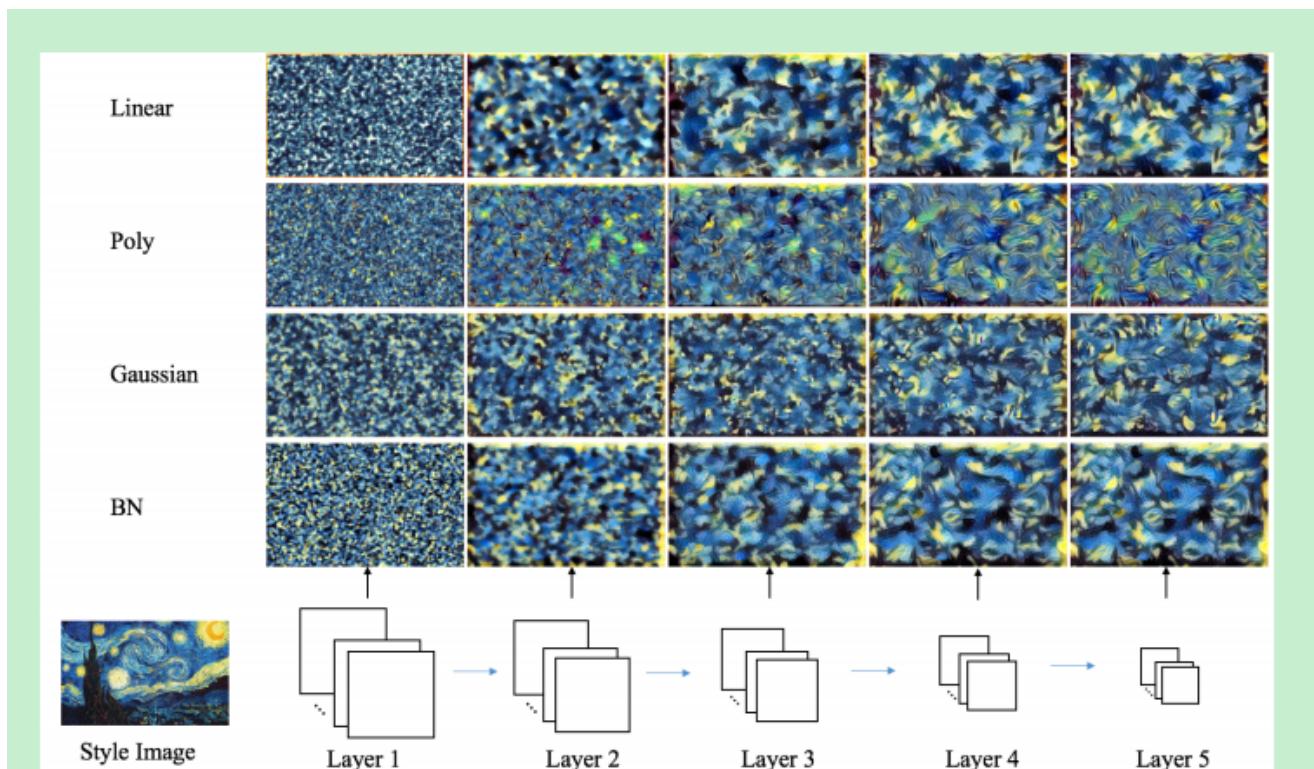


Figure 1: Style reconstructions of different methods in five layers, respectively. Each row corresponds to one method and the reconstruction results are obtained by only using the style loss \mathcal{L}_{style} with $\alpha = 0$. We also reconstruct different style representations in different subsets of layers of VGG network. For example, layer 3 contains the style loss of the first 3 layers ($w_1 = w_2 = w_3 = 1.0$ and $w_4 = w_5 = 0.0$).

由图可知：

1. 不同层捕捉了不同级别的分割：深层纹理相比于浅层有更大的粒度。这个非常合理，因为深层的特征有更大的感受野，因此可以捕捉更加全局的纹理。
2. 不同种风格迁移方法捕捉到的风格是不一样的。

3. 平衡因子的影响

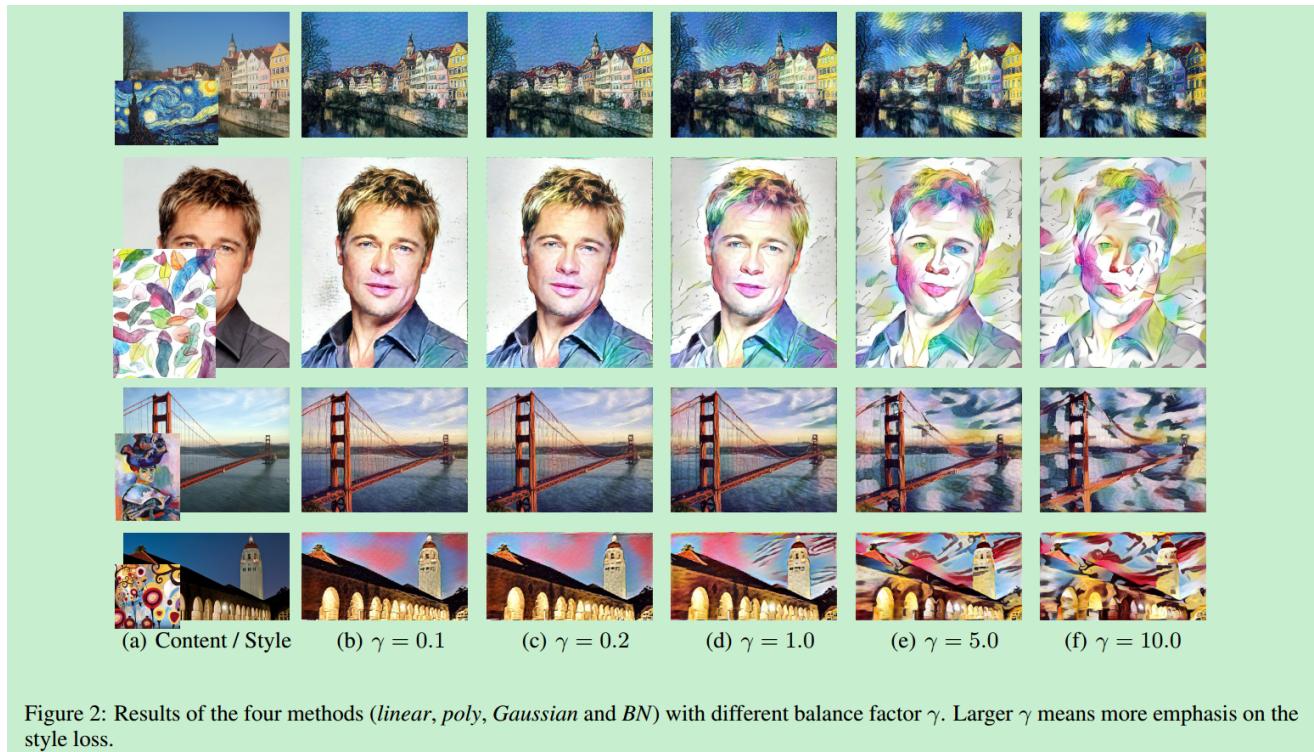


Figure 2: Results of the four methods (*linear*, *poly*, *Gaussian* and *BN*) with different balance factor γ . Larger γ means more emphasis on the style loss.

4. 不同迁移方法的比较



Figure 3: Visual results of several style transfer methods, including *linear*, *poly*, *Gaussian* and *BN*. The balance factors γ in the six examples are 2.0, 2.0, 2.0, 5.0, 5.0 and 5.0, respectively.

5.不同风格迁移方法的融合



Figure 4: Results of two fusion methods: *BN + poly* and *linear + Gaussian*. The top two rows are the results of first fusion method and the bottom two rows correspond to the second one. Each column shows the results of a balance weight between the two methods. γ is set as 5.0.