Implicit Neural Representations for Image Compression

Implicit Neural Representations for Image Compression

Abstract

隐性神经表征（INRs）作为各种 数据类型 的一种新颖而有效的表征而受到关注。最近，先前的工作将 INRs 应用于图像压缩 。这种压缩算法作为一种通用的方法，对任何基于坐标的数据模式都是很有希望的。然而，目前基于 INR 的压缩算法需要在很大程度上改善其率失真性能。这项工作在这个问题上取得了进展。首先，我们为基于 INR 的压缩算法提出了元学习的初始化，从而提高了速率失真性能。此外，它也导致了更快的收敛速度。其次，与之前基于 INR 的压缩工作相比，我们对网络结构进行了简单但非常 有效的 改变。也就是说，我们将 SIREN 网络与位置编码结合起来，提高了速率失真性能。我们对使用 INR 的源压缩的贡献大大超过了先前的工作。我们表明，我们的基于 INR 的压缩算法，与 SIREN 和位置编码相结合的元学习，首次在柯达上以2倍的降低维度超越了 JPEG2000 和速率失真自动编码器，并在全分辨率图像上缩小了差距。为了强调基于 INR 的源压缩的通用性，我们进一步进行了 3D 形状压缩的实验，我们的方法大大超过了 Draco 一种传统压缩算法。

1. Introduction

生活在一个数字化无处不在、重要决策基于大数据分析的世界里，如何有效地存储信息的问题比以往任何时候都更重要。源压缩是以紧凑形式表示数据的总称，它要么保留所有信息（无损压缩），要么牺牲一些信息以获得更小的文件尺寸（有损压缩）。它是解决每天从 互联网 上传、传输和下载的大量图像和视频数据的一个关键组成部分。虽然无损压缩可以说是更理想的，但它有一个基本的理论限制，即香农的熵[47]。因此，有损压缩的目的是将文件的质量与它的大小进行权衡–称为率失真权衡。

除了为特定数据模式（如音频、图像或视频）调整的传统手工设计的算法外，机器学习研究最近通过利用神经网络 的力量开发了有前途的源压缩学习方法。这类方法通常是在著名的自动编码器[28]的基础上，实现其约束版本。这些所谓的速率失真自动编码器（RDAEs）[5,6,37,25] 共同优化了解码数据样本的质量和其编码文件的大小。

这项工作避开了RDAEs的普遍做法，研究了一种新的源压缩范式–特别是专注于图像压缩。最近，隐式神经表征（INRs）作为一种灵活的、多用途的数据表征而受到欢迎，它能够在图像[49]、三维形状[44,49]和场景[40]上产生高保真的样本。一般来说，INRs通过学习网格的坐标和相应的数据值（如RGB值）之间的映射来表示生活在底层规则网格上的数据，甚至被假设为能产生良好的压缩表示[49]。由于它们的通用性和早期利用它们进行压缩的尝试[19,11,20]，INRs表示一种有希望成为通用压缩算法的候选。

目前，基于INR的压缩算法有两个主要挑战：（1）直接的方法甚至难以与最简单的传统算法竞争[19]。(2) 由于INR是通过对特定实例的过度拟合来编码数据的，编码时间被认为是不切实际的。为此，我们做出了两个贡献。首先，我们提出了基于INR的压缩的元学习（meta-learned）。我们利用最近在基于模型预知元学习（MAML）[22]的INR元学习[48,52]方面的进展，找到可以用较少的梯度更新来压缩数据的权重初始化，并产生更好的额定失真性能。其次，我们将SIREN与位置编码结合起来，进行基于INR的压缩，大大改善了速率-失真性能。虽然我们专注于图像，但我们强调，我们提出的方法可以很容易地适应任何基于坐标的数据模式。总的来说，我们引入了一个压缩管道，其性能大大超过了最近提出的COIN[19]，并与传统的 图像压缩算法 相竞争。此外，我们证明了元学习的INR已经超过了JPEG2000和一些下采样图像的RDAEs的性能。最后，我们强调了基于INR的图像压缩的通用性，将我们的方法直接应用于3D数据压缩，我们的表现超过了传统算法Draco。

3. Method

3.1 Background

INR存储基于坐标的数据，如图像、视频和三维形状，将数据表示为从坐标到数值的连续函数。例如，一幅图像是一个水平和垂直坐标（px，py）的函数，并映射到色彩空间（如RGB）中的一个颜色向量：

$$I : (p_x, p_y) \rightarrow (R, G, B) \tag{1}$$

这个映射可以由一个神经网络 f_θ 近似，通常是一个具有参数 θ 的多层感知器（MLP），这样 $I(p_x, p_y) \approx f_\theta(p_x, p_y)$。由于这些函数是连续的，INRs是与分辨率无关的，也就是说，它们可以在归一化范围[-1，1]内的任意坐标上进行评估。为了表达一个基于像素的图像张量x，我们在一个均匀间隔的坐标网格p上评估图像函数，这样 $x = I(p) \in \mathbb{R}^{W \times H \times 3}$，其中

$$\mathbf{p}_{ij} = \left(\frac{2i}{W-1} - 1, \frac{2j}{H-1} - 1 \right) \in [-1, 1]^2 \quad \forall i \in \{0, \ldots, W-1\}, j \in \{0, \ldots, H-1\}. \tag{2}$$

请注意，每个坐标 vector 都是独立映射的：

$$f_\theta(\mathbf{p}) = \begin{bmatrix} f_\theta(\mathbf{p}_{11}) & \cdots & f_\theta(\mathbf{p}_{1H}) \\ \vdots & \ddots & \vdots \\ f_\theta(\mathbf{p}_{W1}) & \cdots & f_\theta(\mathbf{p}_{WH}) \end{bmatrix}. \tag{3}$$

速率失真自动编码器。在学习源压缩中最主要的方法是RDAEs： 编码器网络产生一个压缩的表示，通常称为潜向量z∈R^d，联合训练的解码器网络使用它来重建原始输入。早期的方法通过限制其维度d来加强z的紧凑性[27]。较新的方法通过在损失中加入一个熵估计，即所谓的速率损失，来约束表示。这个反映z的存储要求的速率项，与量化压缩误差的失真项一起被最小化。

3.2 Image Compression using INRs

与RDAE相反，INR将所有信息隐含地存储在网络权重θ中。INR的输入本身，即坐标，不包含任何信息。编码过程等同于训练INR。解码过程相当于将一组权重加载到网络中并在坐标网格上进行评估。我们可以将其总结为：

$$\arg\min_{\theta} \mathcal{L}(\mathbf{x}, f_\theta(\mathbf{p})) = \theta^\star \xrightarrow[\text{transmit } \theta^\star]{} \widehat{\mathbf{x}} = f_{\theta^\star}(\mathbf{p}). \tag{4}$$

因此，我们只需要存储 θ· 来重建原始图像 x 的扭曲版本。通过我们的方法，我们描述了一种寻找 θ· 的方法，以实现紧凑的存储并同时实现良好的重建。

架构。我们使用 SIREN，即一个使用正弦激活的 MLP，频率 ω=30，最初在[49]中提出，最近在图像数据上显示出良好的性能。我们采用了作者建议的初始化方案。由于我们的目标是在多个比特率下评估我们的方法，我们改变了模型的大小以获得速率-失真曲线。我们还提供了关于如何改变模型大小以达到最佳速率-失真性能的分析（见补充材料）和关于INR的结构（见第4.4节）。

输入编码。输入编码将输入坐标转换到一个更高的维度，这已被证明可以改善感知质量[40,53]。值得注意的是，据我们所知，我们是第一个将SIREN与输入编码结合起来的，以前输入编码只用于基于整流线性单元（ReLU）激活函数的INR。我们应用了[40]中提出的位置编码的改编版本，其中我们引入了比例参数σ来调整频率间隔（类似于[53]），并将频率项与原始坐标p连接起来（如SIREN代码库1）：

$$\begin{aligned}\gamma(p) =& (p, \sin(\sigma^0 \pi p), \cos(\sigma^0 \pi p), \dots, \\ & \sin(\sigma^{L-1} \pi p), \cos(\sigma^{L-1} \pi p)).\end{aligned} \tag{5}$$

其中 L 是使用的频率数。我们在第4.4节研究输入编码的影响。

3.3 Compression Pipeline for INRs

本节介绍了我们基于INR的压缩管道。首先，我们描述了我们基于随机初始化INR的基本方法（第3.3节）。然后，我们提出了元学习的初始化，以改善基于INR的压缩的速率-失真性能和编码时间（第3.3节）。图2描述了整个管道，图1显示了更高层次的概述。

Basic Approach using Random Initialization. 第一阶段： 过度拟合。首先，我们在测试时将INR fθ过度拟合到一个数据样本上。这等同于调用其他学习方法的编码器。我们称这一步为过拟合，以强调INR被训练为只代表一个图像。给定一个图像x和一个坐标网格p，我们最小化目标：

$$\arg\min_{\theta} \mathcal{L}_{\text{MSE}}(\mathbf{x}, f_\theta(\mathbf{p})). \tag{6}$$

我们使用平均平方误差（MSE）作为损失函数来衡量地面实况目标和INRs输出的相似性：

$$\mathcal{L}_{\text{MSE}}(\mathbf{x}, \widehat{\mathbf{x}}) = \sum_{i}^{W} \sum_{j}^{H} \frac{\|\mathbf{x}_{ij} - \widehat{\mathbf{x}}_{ij}\|_2^2}{WH}. \tag{7}$$

注意，$x_{ij} \in \mathbb{R}^3$ 是单个像素的颜色向量。

正则化。在图像压缩中，我们的目标是同时最小化失真（例如，MSE）和比特率。由于模型熵不是可微的，我们不能直接在基于梯度的优化中使用它。文献中使用的一种方法是在训练期间使用可微分的熵估计器[2]。然而，我们选择使用一个近似诱导低熵的正则化项。特别是，我们对模型权重采用L1正则化。总的来说，这产生了以下的优化目标：

$$\mathcal{L}(\mathbf{x}, f_\theta(\mathbf{p})) = \mathcal{L}_{\text{MSE}}(\mathbf{x}, f_\theta(\mathbf{p})) + \lambda \|\theta\|_1 \tag{8}$$

其中 λ 决定了诱导稀疏性的 L1 正则化的重要性。我们的正则化术语与[46]中采用的稀疏性损失有关：我们有相同的目标，即限制权重的熵，但是我们将其应用于INR，而他们将其应用于传统的显式解码器。

第二阶段： 量化。通常情况下，过拟合产生的模型权重是单精度浮点数，每个权重需要32位。为了减少内存需求，我们使用人工智能模型效率工具箱（AIMET）1对权重进行量化。我们对每个权重张量采用特定的量化，这样均匀间隔的量化网格被调整到张量的值范围。位宽决定了离散水平的数量，即量化仓的数量。我们根据经验发现，位宽在7-8的范围内会使我们的模型获得最佳的速率-失真性能，如补充文件所示。

第三阶段：量化后优化。量化通过将权重四舍五入到最接近的量化仓来降低模型的性能。我们利用两种方法来减轻这种影响。首先，我们采用AdaRound[42]，这是一个二阶优化方法来决定是向上还是向下舍入一个权重。其核心思想是，传统的最接近四舍五入并不总是最好的选择，正如[42]中所示。随后，我们使用量化感知训练（QAT）对量化后的权重进行微调。这一步骤旨在扭转部分量化误差。量化是不可微分的，因此我们依靠直通估计器（STE）[8]进行梯度计算，基本上绕过了反向传播过程中的量化操作。

第四阶段：熵编码。最后，我们进行熵编码以进一步无损压缩权重。特别是，我们使用二进制算术编码算法来无损地压缩量化的权重。

用于压缩INRs的元学习初始化。直接将INRs应用于压缩有两个严重的限制：首先，它需要在编码步骤中对数据样本从头开始过度拟合模型。其次，它不允许在压缩算法中嵌入归纳偏见（例如，对特定图像分布的了解）。为此，我们应用meta-learning，即模型不可知元学习（MAML）[23]，来学习接近权重值的权重初始化，并包含图像分布的信息。以前关于INR的元学习的工作主要是为了提高收敛速度[52]。学会的初始化θ0据称在权重空间中更接近于最终的INR。我们希望在更新Δθ=θ-θ0比完整的权重张量θ需要更少的存储空间的假设下，利用这一事实进行压缩。因此，我们固定θ0并将其包含在解码器中，这样就足以传输Δθ，或者准确地说，量化的更新Δ˜θ。然后，解码器可以通过计算重建图像：

$$\tilde{\theta} = \theta_0 + \Delta\tilde{\theta}, \quad \widehat{\mathbf{x}} = f_{\tilde{\theta}}(\mathbf{p}). \tag{9}$$

我们希望权重更新Δθ所占据的数值范围明显小于全部权重θ的数值范围。在固定的位宽下，量化仓之间的步长在权重更新的情况下会更小，因此，平均舍入误差也会更小。

请注意，在过度拟合单个图像之前，每个分布D只学习一次初始化。因此，我们把它介绍为第0阶段。第0阶段发生在训练时间，在许多图像上执行，不是推理的一部分。阶段1-4发生在推理时间，目的是压缩单一图像。因此，使用元学习的初始化并不增加推理时间。