Point cloud attribute compression with neural implicit representations

8th INTERNATIONAL WORKSHOP ON ON-BOARD PAYLOAD DATA COMPRESSION (OBPDC 2022)

28-30 SEPTEMBER 2022

POINT CLOUD ATTRIBUTE COMPRESSION WITH NEURAL IMPLCIIT REPRESENTATIONS

Francesca Pistilli[(1)], Diego Valsesia[(1)], Giulia Fracastoro[(1)], Enrico Magli[(1)]

*[(1)]**Politecnico di Torino***
C.so Duca degli Abruzzi 24, Torino, 10129, Italy
Email enrico.magli@polito.it:

ABSTRACT

Point cloud attribute compression is a challenging task due to the irregularity of the point cloud domain. This makes it difficult to extend traditional compression principles like transform coding to such data type, often requiring complex and sophisticated schemes. At the same time, deep learning is gaining popularity as a way to design optimized compression algorithms. Existing end-to-end signal compression schemes using neural networks are largely based on an autoencoder-like structure, where a universal encoding function creates a compact latent space and the signal representation in this space is quantized and stored. In this paper, we follow a different approach by adopting neural implicit representation networks, i.e., neural networks that are queried with a coordinate and returns the signal value at that coordinate. A network of this kind is trained to overfit the signal to be compressed and the neural network itself, in its weights and biases, becomes the compressed representation of the signal. Efficient techniques to quantize neural network weights are then used to limit the rate of the compressed representation. We also show that it is possible to induce prior knowledge about the class of signals of interest via meta-learning techniques, thus providing an initialization value for the network weights. This procedure has a twofold advantage in terms of complexity and compression efficiency. In particular, it allows to finetune the network for the representation of the specific signal of interest with a small number of iterations, limiting encoding complexity. Moreover, the weights can be encoded differentially with respect to such initialization to achieve greater rate-distortion efficiency. Preliminary experiments show that the proposed method is competitive with the latest G-PCC MPEG standard for point cloud attribute compression, and outperforms RAHT, a recent state-of-the-art method.

INTRODUCTION

The rise of neural networks due to their exceptional performance on computer vision tasks and beyond has stimulated research on their use in domains traditionally focused on model-based techniques rather than data-driven methods. In particular, data compression has been traditionally tackled by means of transform coding or predictive coding techniques in order to capture the complex correlation patters that real signals exhibit.
Recently, convolutional neural networks (CNNs) have shown great promise in image, video and point cloud compression [1,2,3], either as supplements to existing codecs by replacing specific modules or by entirely replacing the compression pipeline. This is the case of the so called end-to-end techniques where the whole codec is a neural network which can be fully optimized to achieve the best rate-distortion performance on a class of data.
The promising performance of such techniques is linked to the exception representation learning capabilities of convolutional neural networks. In essence, via suitable training data, they are able to learn compact domains which capture the most salient features.

Indeed, the dominating approach in the literature is the use of architectures in the form of auto-encoders. In this framework, an encoder neural network maps the input data into a compact latent space, such as a short vector, which quantized and entropy coded. A decoder neural network maps such latent vectors back into an estimate of the original data. The architecture can be simply trained by means of a loss function which measures the distortion achieved by the auto-encoding process, with some technical issues around the non-differentiability of the quantization operation which are currently addressed in a number of ways.

The auto-encoder paradigm is based on the idea that a universal encoder function can be learned to generate a compact space in which any input signal can be faithfully represented. However, care must placed on the implementation of this idea. In fact, practically learning a good universal encoder hinges on collecting large quantities of data which are as faithful as possible to the data that will be processed during inference. There is also a potential lack of flexibility in this

approach when input signals with widely different characteristics (e.g. different resolutions) must be handled. Moreover, developing good encoders for certain signals might be challenging. The is the case of point cloud data which are supported on an irregular domain an cannot use conventional designs for grid, like CNNs and must resort to more complex, computationally expensive and tricky to optimize designs like graph neural networks.

In this paper, we present a novel paradigm to use neural networks for signal compression that is alternative to the dominant autoencoding model. The idea is based on implicit neural representations, which are simple neural networks that take as input a single coordinate value from the signal domain (e.g., a pixel position in an image) and return the value of the signal at that coordinate (e.g., that pixel brightness). In this framework, the weights of the neural network are overfit on a single signal of interest and the neural network in its weights and biases becomes the signal itself. This means that if we could design a network that uses a smaller number of weights than the signal coefficients, compression would be achieved.

In this paper we show how to design a signal compressor based on neural implicit representations, propose a method to efficiently encode the network weights and provide prior knowledge to the network about the set of signals of interest to reduce the computational complexity of the compression procedure as well as improve rate-distortion performance.

Preliminary results are presented on the task of point cloud attribute compression, which is challenging for traditional encoders to the irregular nature of point clouds.

COMPRESSION WITH NEURAL IMPLICIT REPRESENTATIONS

Neural implicit representations have recently enjoyed great success in the computer graphics community [4] where they are used to represent the lightfield of a scene and render images from novel viewpoints. A neural implicit representation is a simple multilayer perceptron (MLP, a neural network with multiple fully-connected layers) that takes as input a coordinate **x** from the D-dimensional signal domain and returns the value of the signal at that coordinate f(**x**). In the point cloud attribute compression sample application that we discuss in the experiments, **x** is the 3D position of a point and f(**x**) is the attribute (e.g. color) value at that point. Notice that the input is a single coordinate, so that weights and biases of the network are the same for any coordinate that is processed. The network overfits a specific signal f, so that it *becomes* the signal. The rate of the signal representation is entirely determined by how many weights the network has and how efficiently we are able to store them. The distortion is determined by how faithful the network output is to the original signal.

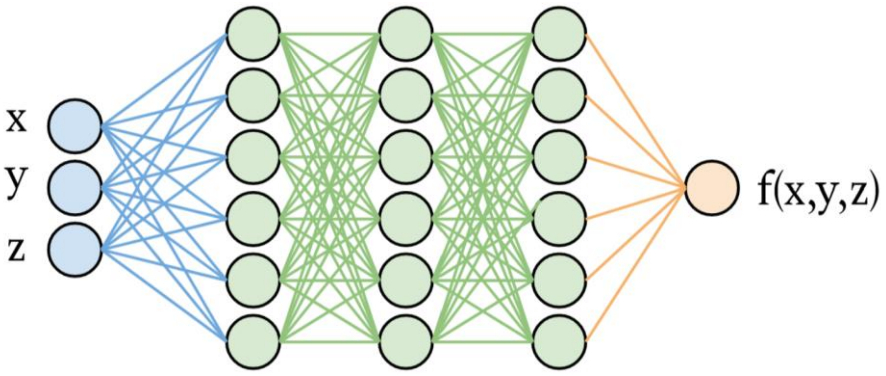

Figure 1. A neural implicit representation network maps a coordinate from the signal domain into the signal value at that coordinate.

Only recently these kind of networks achieved interesting performance due to the realization that special care must be placed in the design of the first layer which maps the low-dimensional coordinate value into a high-dimensional feature space. This operation is referred to as positional encoding, and the most successful positional encodings currently known is the use of sinusoid activation functions as in SIRENs [5] (actually, the authors propose the use of sinusoidal activations for all the layers in the network due to stable behavior of their derivatives of any order) or Fourier embeddings [6]. These solutions for positional encoding allow to better represent the high-frequency content of the signal.

Therefore, in order to compress a signal via neural implicit representations, the following operations must be performed:

1. Design an MLP with a suitable positional encoding mechanism, number of layers and number of neurons in the hidden layers. Scaling laws (is it better to use more layers or more neurons?) for the these models are still unclear at this point.

2. Define a suitable regression loss to measure distortion (e.g., the mean squared error) and use it to train the network.

$$\hat{\theta} = argmin_{\theta} \sum_{x} \|f_{\theta}(\boldsymbol{x}) - f(\boldsymbol{x})\|^2$$

   For signals with very large domains, the set of all coordinates can be split into minibatches with random subsets of coordinates.
3. Represent the weights and biases of the network in a compact way. This involves techniques like sparsification and quantization. We notice that implicit representation network are much more sensitive to weight quantization than their classification counterparts and significant losses in rate-distortion performance appear for too aggressive quantization.
4. Use an entropy encoder to on the quantized weights and biases and save any side information like sparsification patterns, etc.

Decoding the compressed signal just amounts to a forward pass through the network for all the coordinates of interest.

A naïve usage of the proposed framework would require training the neural network from random initialization for every signal to be compressed. However, this has a number of limitations and it is possible to devise a pretraining procedure which serves the both the purpose of improving the rate-distortion performance of the method and reducing the computational complexity.
A randomly initialized neural implicit representation neuron does not possess any prior knowledge about our signals of interest. For example, it does not know that locality is important in representing images. We therefore would like to instil some general knowledge about the class of signals we want to compress. This is possible by pretraining the network with a meta-learning procedure. This procedure simply amounts to training the network to represent not one single signal, but an entire dataset of signals that are representative of the ones of interest. This training simply iteratively updates the weights to represent a randomly drawn minibatch of signals from the dataset. We notice that this procedure will not converge to a good representation of any of the signals in the dataset, but rather to a sort of "average" signal. In doing so, however, the network layers learn extractors of features that can be found in the signals of interest, building some form of prior.

Once we need to compress a signal, we can finetune the pretrained network on the single target signal. Thanks to the pretraining step, this typically will converge to the same distortion of the naïve procedure but in a significantly smaller number of iterations. As an example, in the experiments on point cloud attribute compression we observe converge in about 10000 iterations instead of several hundreds of thousands, making the finetuning (compression) more feasible on hardware with limited capabilities instead of powerful GPUs.

Concerning the rate of the compressed representation, we observe that the finetuned values of the network weights are highly correlated with the starting values provided by pretraining. Since pretraining is universal, it could performed once and the pretrained weight values provided as part of a standard specification. It would just then suffice to store the difference between the finetuned values and the pretrained values, which has a significantly smaller dynamic range and allows for more faithful representation of the weights.

EXPERIMENTS ON POINT CLOUD ATTRIBUTE COMPRESSION

In this section, we present the application of the proposed compression framework to a notoriously challenging task: point cloud attribute compression. This task is challenging because the attributes are supported on an irregular domain, due to the points of the cloud taking arbitrary positions in 3D space, instead of aligning on a grid like the pixels of an image.
In the following experiments, we consider a SIREN neural network for our implicit neural representations. We use the Microsoft Voxelized Upper Bodies [7] and the 8i Voxelized Full Bodies [8] datasets for our experiments. In particular, we pretrain the network using the Microsoft Voxelized Upper Bodies, MAML [9] as meta-learning algorithm with an outer loop over the dataset which optimizes the weights using Adam with 10^{-5} learning rate and inner loop over coordinates using Stochastic Gradient Descent with 10^{-2} learning rate. The number of outer iterations is 1000 and the number of inner iterations is 10. Finetuning is performed on the signals to be compressed from the 8i Voxelized Full Bodies. Notice that the two datasets are strictly disjoint in content. For finetuning we use Adam optimizer with 10^{-5} learning rate for up to 20000 iterations. The loss function to be minimized the MSE in the YUV color space:

$$L = \alpha MSE_Y + \beta MSE_U + \gamma MSE_V$$

where $\alpha = 0.6, \beta = 0.2, \gamma = 0.2$ modulate the relative contributions of luminance and chrominance.
After finetuning, weights and biases are differentially encoded with respect to the pretrained values. The differences are quantized with uniform scalar quantizer, whose quantization step size is optimized layer-by-layer. Finally, the quantized differences are entropy coded with an arithmetic encoder.

First, we analyse how the pretained values of the weights can serve as a good predictor of the final value after finetuning. Figure 2 shows that the dynamic range is significantly compressed by considering the differences with respect to the pretrained value rather than the raw weight value.

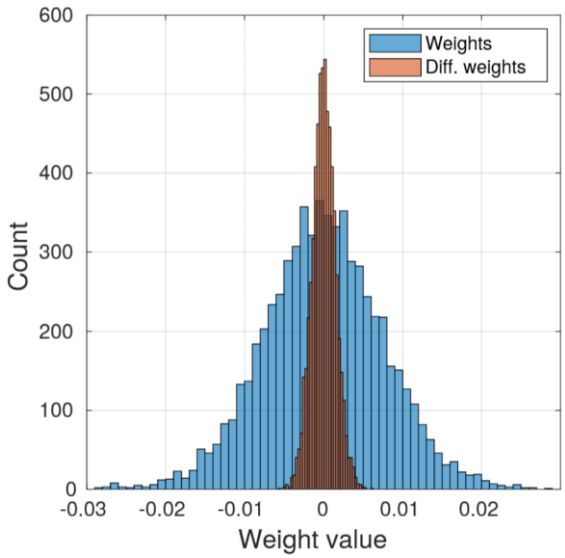

Figure 2. Distribution of network weights and differences between finetuned and pretrained weights.

Then, we look at the rate-distortion performance obtained by compressing the attributes of the *Loot_vox10_1200, Longdress_vox10_1300, Redandblack_vox10_1550,* and *Soldier_vox10_0690* point clouds from the test set. Figure 3 shows the comparison with a recent state-of-the-art algorithm called RAHT [10] and the latest MPEG G-PCC standard (v12.0). The rate-distortion curve for the proposed method (NIC – Neural Implicit Compression) is obtained as the envelop of the curves spanning the design degrees of freedom, namely number of layers, neurons and quantization step sizes. We restricted our choice to networks with 60,80 or 130 neurons per hidden layer and 5,7, or 9 layers. Quantization step sizes vary corresponding to a number of levels ranging from 2^2 to 2^{12}. We notice that the proposed method outperforms RAHT and is very close to the highly optimized MPEG standard.

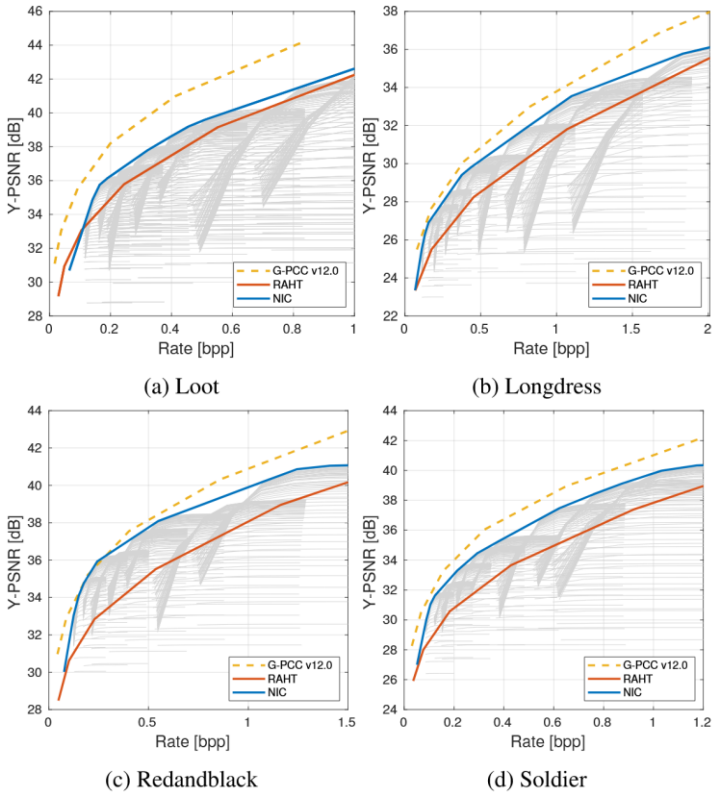

Figure 3Rate-distortion performance on Y-PSNR.

Table 1 also reports the BD-Rate results over total PSNR for the proposed method against RAHT.

Table 1. BD-Rate over total PSNR versus RAHT.

Loot	Longdress	Redandblack	Soldier
-10.23%	-38.88%	-22.10%	-33.39%

CONCLUSIONS

We introduced a novel framework for compression using neural networks and tested its preliminary performance on the challenging task of point cloud attribute compression. The framework has potential to be competitive with the dominant approach of auto-encoders and its per-signal optimization could outperform it if domain gaps exist between training and inference data in the autoencoding framework. Further work is needed to optimally design and compress neural implicit representations, especially for images where it is hard to exploit strong priors like conventional methods do.

REFERENCES

[1] Johannes Balle, Valero Laparra, and Eero P Simoncelli, "End-to-end optimized image compression," in International Conference on Learning Representations, 2016

[2] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao, "Dvc: An end-to-end deep video compression framework," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019

[3] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux, "Folding-based compression of point cloud attributes," in 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 3309–3313

[4] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in European conference on computer vision. Springer, 2020, pp. 405–421

[5] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein, "Implicit neural representations with periodic activation functions," Advances in Neural Information Processing Systems, vol. 33, 2020

[6] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," Advances in Neural Information Processing Systems, vol. 33, 2020

[7] C. Loop, Q. Cai, S.O. Escolano, and Philip A. Chou, "Microsoft voxelized upper bodies – a voxelized point cloud dataset," in ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012, May 2016.

[8] Eugene d'Eon, Bob Harrison, Taos Myers, and Philip A. Chou, "8i voxelized full bodies - a voxelized point cloud dataset," in ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006, January 2017.

[9] Alex Nichol, Joshua Achiam, and J. Schulman, "On first-order meta-learning algorithms," ArXiv, vol. abs/1803.02999, 2018

[10] Ricardo L. de Queiroz and Philip A. Chou, "Compression of 3d point clouds using a region-adaptive hierarchical transform," IEEE Transactions on Image Processing, vol. 25, no. 8, pp. 3947–3956, 2016.