"Transformers as Meta-Learners for Implicit Neural Representations" paper review

Nikita Semenov
Department of mechanics and mathematics
Moscow State University
nikita2014semenov@gmail.com

Abstract

Implicit Neural Representation (INR) is drowing several benefits over discrete representations, and many scientists work to improve this techique. Common approach implies applying meta-learning, and in considered article authors suggest to use transformers instead traditional gradient descent (like MAML) to perform weights fine-tune. At [1] was shown that such trick has significant potential in improving speed and generalization of learner and, as shown below, can be much more scalable. But despite all advantages, this approach will face with troubles if target function (which bulding by hypernetwork) doesn't have an appropriate parameterization. This review highlights perspective features of transformers as meta-learners, proposes new parameterization to target function and gives some reason why it must synergizes with transformers and improve generalization and performance of building INR.

1 Transformers as meta-learner

Interest to INR has significantly grown in recent years. Early works like [3] propose the main idea, namely using neural network for memorizing continuous objects like pictures or scenes. This approach implies fitting function f_θ that map any point of object to their features (for example, point of picture to its RGB representation, Figure 1), and [2], [3] demonstrates memory efficiency of INR compared with traditional pointwise representation. Moreover, INR allows raising resolution as far as need with no memory cost (supposed enough capacity of representation function).

But training neural function from scratch is very unefficient by time and demonstrate sufficient troubles with generalization. Brilliant solution of this problem was proposed at several articles (for example, [2]) and is contained in applying meta-learning to find good initial weights with few-shot fine-tuning in inference. Such weights contain information from many objects of train set, that allows to fit fast on any current object and also to learn deep abstract features, which is out of reach without information from many examples.

In this context, developing new methods to perform meta-learning to INR seems like a perspective task. At [1] instead classical gradient descent transformers are used to perform fine-tuning learnable weights θ. There is the good decision because of neural network optimizers are less susceptible to noise gradients problem and slow converges problem and also demonstrate robustness to learning rate size [6]. Transformer like a hypernetwork also has several advantages: residual connections between encoder and decoder blocks reflect a gradient descent idea of consecutive changing of θ with respect to current context (which much more efficiency taken into consideration by multi-head attention then

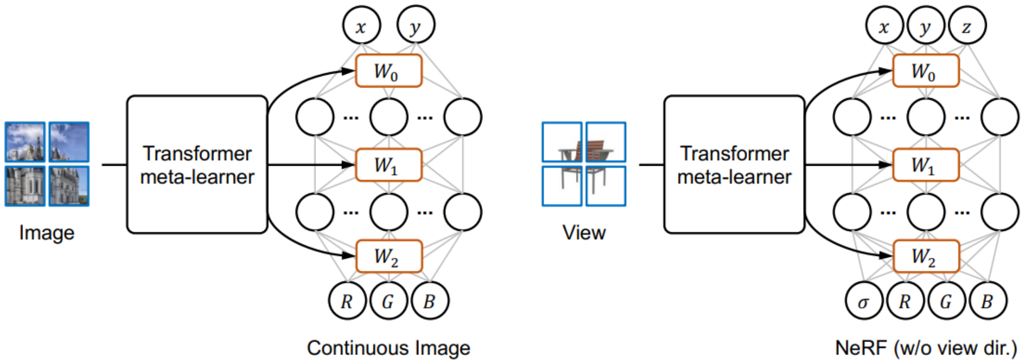

Figure 1: Meta-learning of continuous representation.

by gradients of loss), as well as transformer architecture isn't depending of input sequence length what makes it well-scalable solution. Weight grouping technique (Figure 2) proposed in [1] also growing scalability at several times, so it can be argued that transformers as meta-learners solve scalability and generalization problem of INR.

1.1 Technical problems

Nevertheless, some technical details in paper are not clear. First omission is that authors declared number of residual steps in weights evolution like a learnable parameter, but in transformer this value is determined by number of encoder and decoder blocks. Since that block is learnable structure, its output may has a small norm and not significantly affects on θ, but gradients around extremum also have a small norm. So, learnable number of steps can't be an advantages of proposed work as well as isn't sufficient advantages in fact.

There is another crucial parameter of method which didn't highlight at the paper. Authors didn't said anything about inference time of transformer relative by gradient descent. It potentially might be another benefit of transformer, but to achieve this goal now some experiments additionally required. Anyway, timecost of proposed solution is still interesting and non-estimated parameter, as well as comparison it with another one at gradient descent.

Proposed solution also allows many improvements with well-known tricks like dynamical neuron wirings in target function f_θ (mechanism like in [4]), progressive capacity of f_θ via consequently

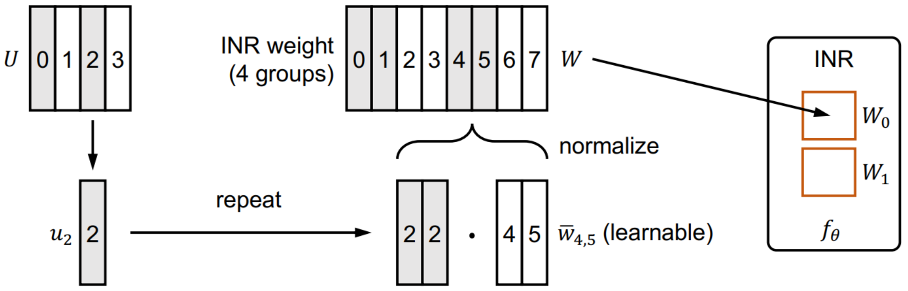

Figure 2: **Weight Grouping.** Columns in weight matrix W are divided into groups, each group can be generated by a single vector. $\overline{w}_i$ are learnable vectors assigned for every column in W, which are independent of the input observations, U is the hypernetwork output.

adding new initial state's tokens (mechanism like in [5]) and etc. But this improvements aren't domain specifically, they able to applying to any network and are objects of independent researches. There is another line of research which isn't considered at paper, takes shape by INR domain and can improve both quality and memory efficiency of representation. It's about using different parameterization of f_θ, which can be more adequate to INR task.

2 Motivation to another parameterization

Since f_θ match points of object with its features, f_θ should have some specific properties which conditioned by nature of object. For example, if we deal with image, we can expect that f_θ will be piecewise smooth with discontinuity at boarders. MLP used in original paper is better modeling continuous functions and hasn't good estimation of convergence to smooth functions speed. This architecture has proven itself at general tasks where target function hasn't any known properties, but since aim is at construct INR of well-known object, it's good way to use peculiarities of domain. Below one parameterization is proposed to solve discontinuous and convergence speed problems.

Denote $f_\theta(x) = \sum\limits_{k=1}^{n} \mathbb{I}(\theta, x) f_k(\theta, x)$, where $f_\theta, f_k : \mathbb{R}^d \to \mathbb{R}$, $\mathbb{I}(\theta, x)$ is indicator of some set in $\mathbb{R}^d$, parameterized by θ. Also denote $f_k = \sum\limits_{j=1}^{n_k} \theta_j^k e^{i(w_k x)}$, $w_k \in \mathbb{Z}_+^d$. Simply saying, denote f_θ as sum of indicators of learnable set multiplied by multidimensional fourier expansion of learnable function on this set. Such decomposition is able to learn circuits of object by indicators and has good convergence speed (since fourier coefficients tends to zero proportional of smoothness of function and value of derivatives). This method will work bad at high-dimensional objects because of number of fourier coefficients grows exponentially with growing dimensions, but at $\mathbb{R}^2$ 10000 parameters determine decomposition to hundredth member of series by each dimension, and in smooth case it's an extremely high accuracy. So, good parameterization might sufficiently improve quality and memory efficiency of representation.

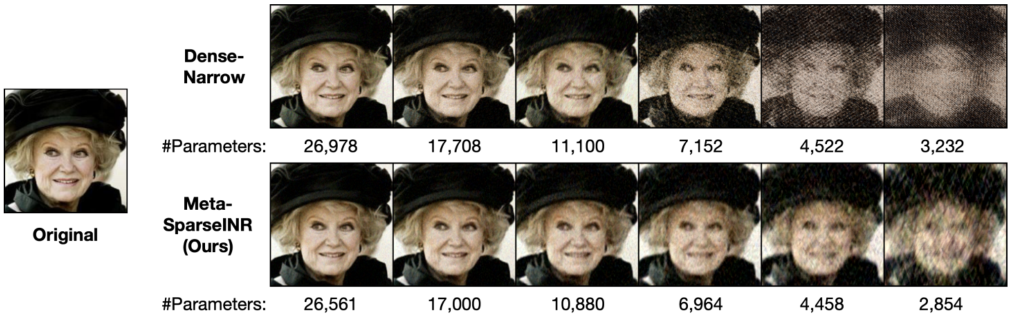

Figure 3: Necessary number of parameters for good quality in solutions proposed before. Using fourier expansion might sufficiently decrease it from Meta-SparseINR values.

3 Required research

Obviously, there are many good parameterizations for f_θ which exploit object structure, fourier decomposition is just one of them. So, there are several research tasks, which can help to build effective and performance solution at INR task:

- Finding crucial properties of potential f_θ.
- Developing parameterizations of f_θ which take into account that properties.

- Testing that parameterizations by number of parameters, inference time, max PSNR, mean PSNR (generalization ability).
- Comparison between proposed parameterisations and finding pros and cons of every one.

First two steps are most important and creative, right here main results can be achieved. Some properties of target function can be delivered by theoretical analysis and comprehension of objects nature (like discontinuous, smoothness, derivative properties and etc.), these ones better find and prove with mathematics. Another ones can be surprisingly found at numerical experiments and modeling (for example, manually construction functions for several objects, generalization properties of them and etc.). As well as finding useful properties, there is complex challenge to develop parameterization which effective exploits its. Classical solution for this task is applying existed parameterizations which are used at mathematics and deep learning, but since we deal with, generally speaking, unknown and bad-determine class of functions, proposed solution must be also robust to target function with a bit different structure.

Last two steps are much more technical then previous ones, so the only question is "how to evaluate solution?". In this work four metrics are proposed, because number of parameters and inference time determine respectively memory and time cost of algorithm, which are classical measures of efficiency, and another two values describe quality of algorithm's output. Max PSNR well characterizes performance in case of the best mapping between properties of real function and used parameterization, whereas mean PSNR well describes the generalization ability, i.e. robustness to variation of real function properties.

4 Summary

Using transformers as meta-learners is extraordinary solution which may be widely applied, but proposed in original article method can be upgraded both at technical and ideological aspects. Useful tricks are described at many works like [4] and [5], as well as research area of this solution must lie at developing performance and compact parameterization of f_θ. Such task is domain-dependent and that's why is the most interested with respect to results.

References

[1] Yinbo Chen & Xiaolong Wang (2022) Transformers as Meta-Learners for Implicit Neural Representations.

[2] Jaeho Lee, Jihoon Tack, Namhoon Lee & Jinwoo Shin (2021) Meta-learning Sparse Implicit Neural Representations.

[3] Emilien Dupont, Adam Golinski, Milad Alizadeh, Yee Whye Teh & Arnaud Doucet (2021) COIN: COmpression with Implicit Neural representations.

[4] Mitchell Wortsman, Ali Farhadi & Mohammad Rastegari (2019) Discovering Neural Wirings.

[5] Tero Karras, Timo Aila, Samuli Laine & Jaakko Lehtinen (2018) Progressive Growing of GANs for Improved Quality, Stability, and Variation.

[6] Olga Wichrowska, Niru Maheswaranathan, Matthew W. Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando de Freitas & Jascha Sohl-Dickstein (2017) Learned Optimizers that Scale and Generalize.