Spatial Function: Scaling Function to ImageNet Classification and Generation

Пискалов Дмитрий БПМИ 212

Вызов

Традиционно данные представляются массивами/тензорами.

Однако удобно ли это всегда? Подумаем для

- гладких объектов в 3D
- изображений
- звука

Массивы они дискретны, а некоторые объекты *на самом деле* непрерывны.

Тогда такие данные удобно ассоциировать с непрерывной функцией.

Как описать такие функции?

Implicit neural representation (INR) — это семейство параметризованных функций.

$$f_\theta : \mathcal{X} \to \mathcal{F}$$

Где $\mathbf{x} \in \mathcal{X}$ — это координата (например пикселя в изображении), а $\mathbf{f} \in \mathcal{F}$ — это характеристика соответствующая координате (например цвет)

Reconstruction loss

Для объекта $\mathcal{I}$ — это index set объекта (например индексное множество для итерации по всем пикселям).

$$\min_{\theta} \mathcal{L}(f_\theta, \{\mathbf{x}_i, \mathbf{f}_i\}_{i \in \mathcal{I}}) = \min_{\theta} \sum_{i \in \mathcal{I}} \| f_\theta(\mathbf{x}_i) - \mathbf{f}_i \|_2^2.$$

$\mathbf{x}_i$ — непосредственно сам пиксель

$\mathbf{f}_i$ — цвет этого пикселя

(на $\mathcal{F}$ определено расстояние)

Определения

Для элемент датасета соответствующая ему INR — это *functa*.

Можем теперь заменить элементы нейросетями (элементами семейства параметризованных функций)

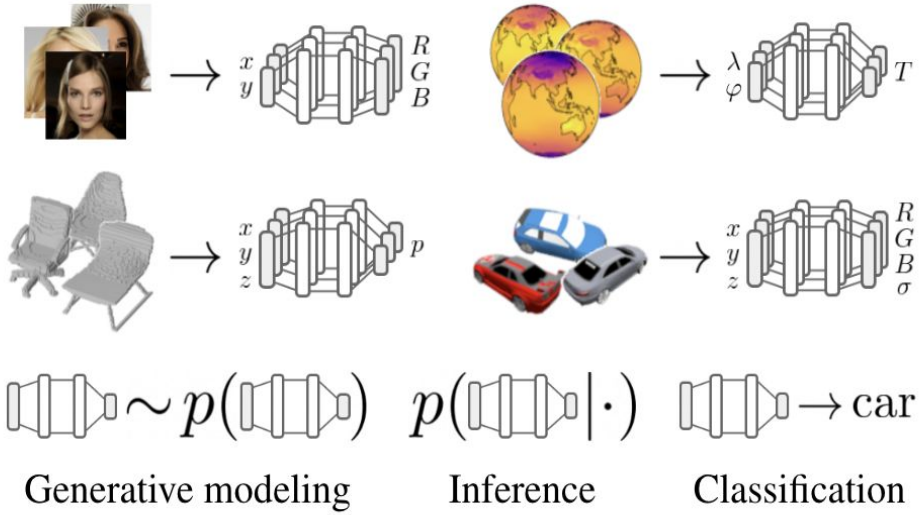
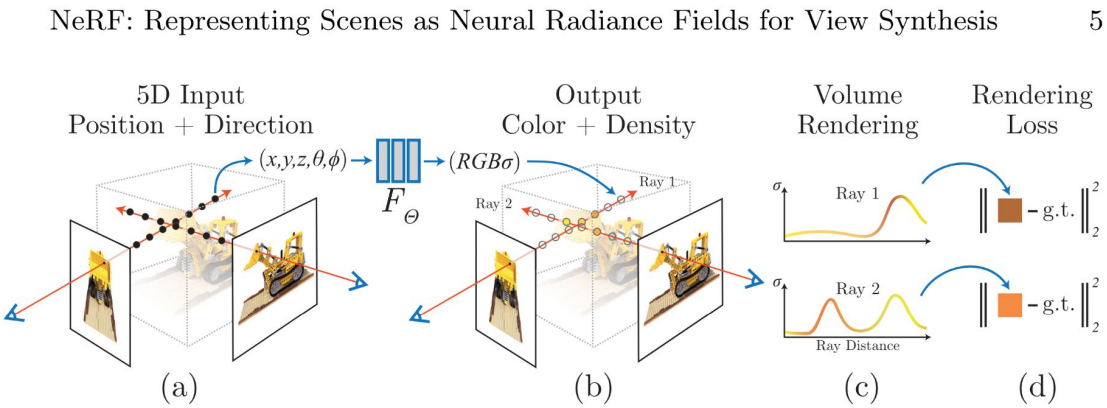


Figure 1. We convert array data into functional data parameterized by neural networks, termed *functa*, and treat these as data points for various downstream machine learning tasks.

Можно вспомнить NeRF. Тогда index set — это индекс сет множества данных картинок.



Преимущества

- зачастую выгоднее параметризовать функции, чем хранить значения во многих точках
- все объекты задаются элементами **одного** однородного множества $\theta \in \Theta$ (размерности картинок же разные)
- некоторые объекты вообще не дискретные сами по себе, их понятно лучше хранить как *functa*

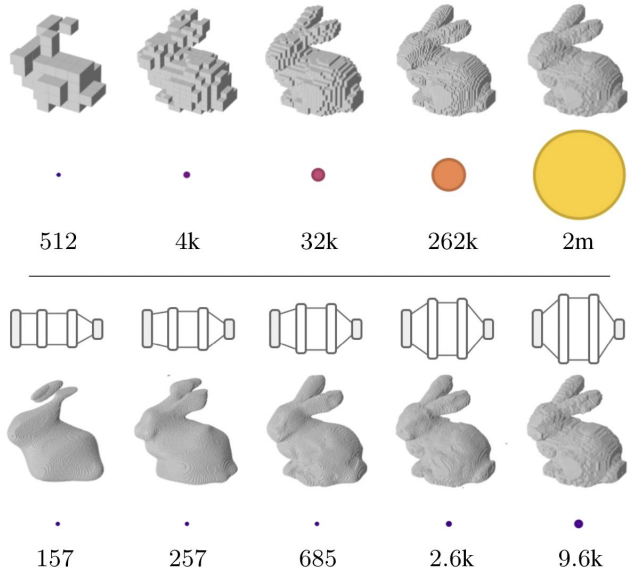


Figure 2. Functa scale much more gracefully with resolution than array representations. Circle area reflects the numerical size of the array (top) / function (bottom). See Appendix A.9 for details.

Вызовы

Преобразование Dataset ⟶ Functaset может быть проблемным

- долгое обучение
- параметризация может быть слишком тяжелой (если это нейросеть)

Решение

Зафиксируем *shared* нейросеть изначальную для всего датасета.

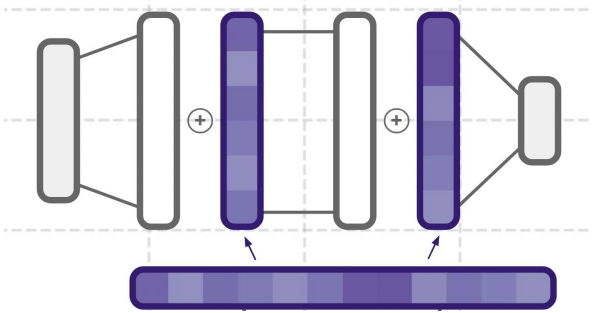
Для каждого элемента будем чуть-чуть ее модифицировать. Но как это реализовать?

Shift modulations

Модификацией shared сети будут прибавление векторов к выходу на каждом слое.

Совокупность этих векторов — это *shift modulations.*

Но shift modulations это набор векторов, хотелось бы получить единый latent vector.

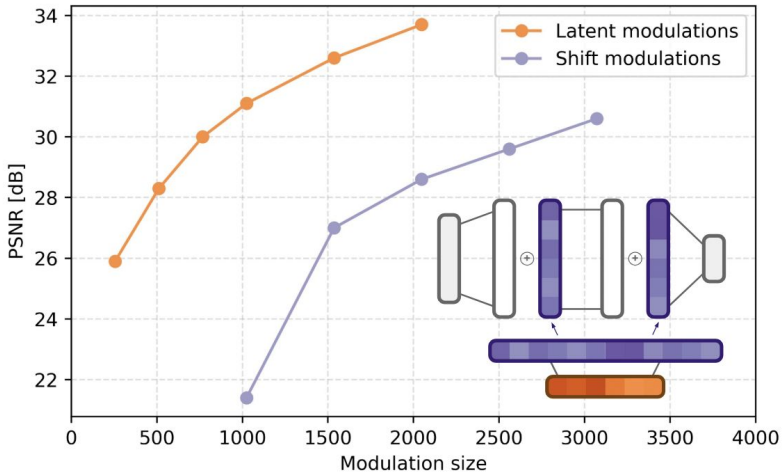


Latent modulation

Параметризация через shift modulations займет все равно много параметров.

$$\text{shift modulations} = \lambda(\textit{latent modulation})$$

where λ is a linear map.



Осталось найти shared сеть и latent modulations быстро.

Поиск shared сети

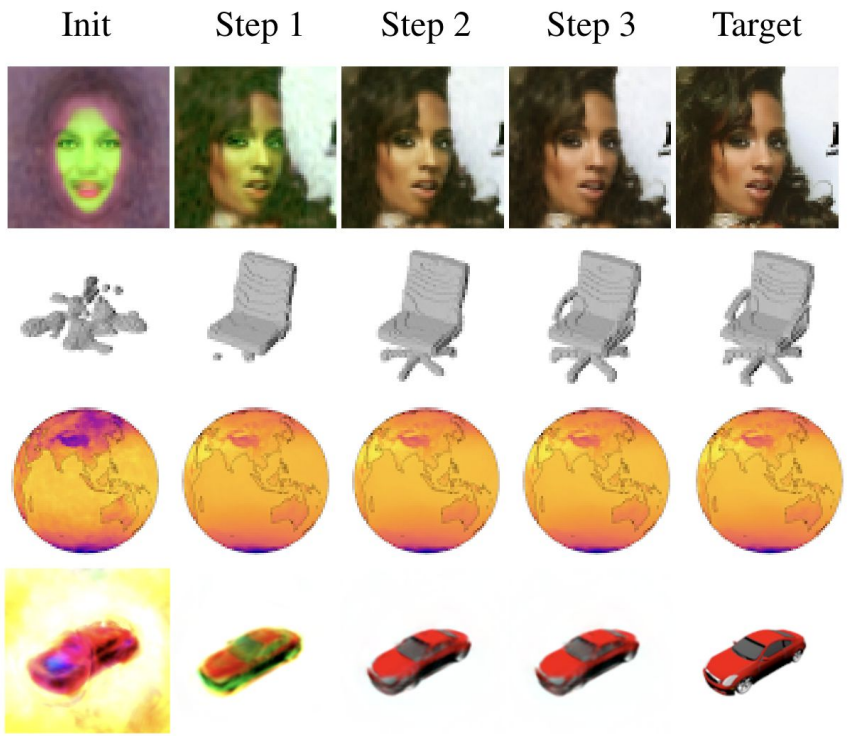
Algorithm 1 Meta-learning functa

Randomly initialize shared base network θ
while not done **do**
  Sample batch $\mathcal{B}$ of data $\{\{\mathbf{x}_i^{(j)}, \mathbf{f}_i^{(j)}\}_{i\in\mathcal{I}}\}_{j\in\mathcal{B}}$
  Set batch modulations to zero $\phi_j \leftarrow 0\ \forall j \in \mathcal{B}$
  for all step $\in \{1, \ldots, N_{inner}\}$ and $j \in \mathcal{B}$ **do**
    $\phi_j \leftarrow \phi_j - \epsilon \nabla_\phi \mathcal{L}(f_{\theta,\phi}, \{\mathbf{x}_i^{(j)}, \mathbf{f}_i^{(j)}\}_{i\in\mathcal{I}})|_{\phi=\phi_j}$
  end for
  $\theta \leftarrow \theta - \epsilon' \frac{1}{|\mathcal{B}|} \sum_{j\in\mathcal{B}} \nabla_\theta \mathcal{L}(f_{\theta,\phi}, \{\mathbf{x}_i^{(j)}, \mathbf{f}_i^{(j)}\}_{i\in\mathcal{I}})|_{\phi=\phi_j}$
end while

Получение latent modulations

Делаем три градиентных шага, минимизируя reconstruction loss.

Как мы и делали во внутреннем цикле в алгоритме для shared сети.



Dataset, array size	Split	Modulation dimensionality 64	128	256	512	1024
ShapeNet Chairs, 64^3	Train	99.43	99.49	99.49	99.51	99.53
	Test	99.11	99.28	99.38	99.46	99.51
ShapeNet 10 Classes, 64^3	Train	99.36	99.44	99.47	99.52	99.56
	Test	99.30	99.40	99.44	99.50	99.55
CelebA-HQ, 64×64	Train	22.2	24.2	26.6	29.7	32.4
	Test	21.6	23.5	25.6	28.0	30.7
SRN Cars, 128×128	Train	24.3	24.2	24.6	24.6	24.4
	Test	22.4	23.0	23.1	23.2	23.1
ERA5, 181×360	Train	43.2	43.7	43.8	44.0	44.1
	Test	43.2	43.6	43.8	43.9	44.0

Table 1. Mean reconstruction of modulations across each dataset vs modulation size. Metric is voxel accuracy (%) for ShapeNet and PSNR (dB) for the rest. See Appendix A.3 for details on metric.

CLASSIFIER	TEST ACCURACY	n_{PARAMS}
MLP ON FUNCTA	$93.6 \pm 0.1\%$	83K
3D CNN	$93.3 \pm 0.3\%$	550K

Table 2. Classification accuracies and parameter count for MLP on functa vs 3D CNN on array data for ShapeNet 10 Classes, 64^3.

Ours - *FID: 80.3*

π-GAN - *FID: 36.7*

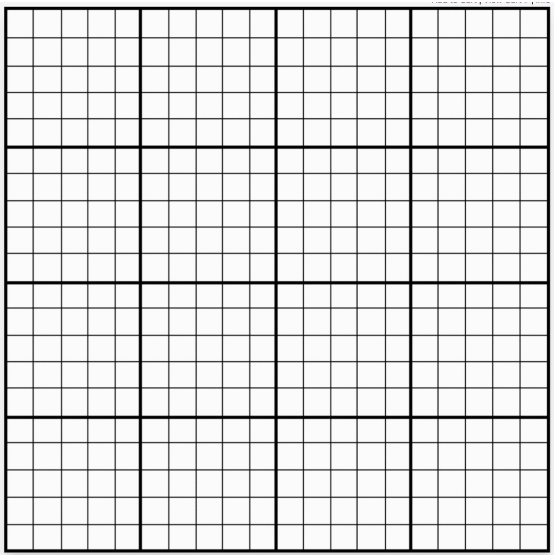
На больших по размеру картинках и привычных датасетах результаты не впечатляют

latent dim	Test PSNR	top-1 acc	FID (uncond)
256	27.6 dB	66.7%	**78.2**
512	31.9 dB	**68.3**%	96.1
1024	**38.1** dB	66.7%	134.8

Table 1: CIFAR-10 functa results

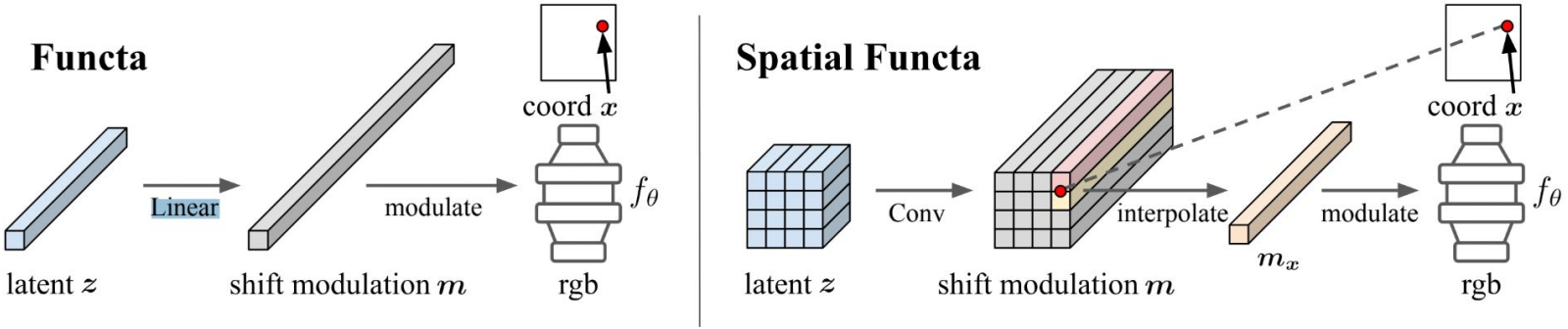
Spatial functa for images

- Ничего не меняется
- Одна shared сеть
- Но latent modulation вектор и shift modulation вектор зависит от участка (патча) картинки



Spatial functa for images

- Картинка делится на s x s патчей размера d/s x d/s
- Будет s x s latent modulation векторов для каждого патча
- Это будет *spatial latent modulation* тензор размера s x s x c
- *Spatial shift modulation* тензор размера s x s x C



Вспоминаем

$$spatial\ shift\ modulations = \lambda(\text{spatial latent modulation})$$

where λ is conv 1x1 with in_channels $= c$, out_channels $= C$.

- Алгоритм поиска shared сети остается тем же
- Для каждого элемента оптимизируем spatial latent modulation тензор
- Для предсказания spatial latent modulation тензоры обрабатываются UNet.

Model	Input shape	Test PSNR ↑	Top-1 acc ↑	FID (cond) ↓
Spatial Functa (1-NN)	$8 \times 8 \times 256$	28.3dB	76.6%	17.9
	$8 \times 8 \times 512$	30.6dB	76.5%	23.5
	$8 \times 8 \times 1024$	25.7dB	76.5%	-
	$16 \times 16 \times 64$	28.9dB	80.4%	12.5
	$16 \times 16 \times 128$	37.8dB	**80.7%**	10.5
	$16 \times 16 \times 256$	37.2dB	80.6%	11.7
	$32 \times 32 \times 16$	28.6dB	-	12.4
	$32 \times 32 \times 32$	31.7dB	-	10.5
	$32 \times 32 \times 64$	37.7dB	-	8.8
	$32 \times 32 \times 64$*	**38.4 dB**	-	**8.5**
ViT-B/16	$224 \times 224 \times 3$	-	79.8%	-
	$384 \times 384 \times 3$	-	**81.6%**	-
LDM-8-G	$32 \times 32 \times 4$ (14 bits)	23.1dB	-	7.8
LDM-4-G	$64 \times 64 \times 3$ (13 bits)	27.4dB	-	**3.6**

Table 3: ImageNet classification and diffusion results. *Indicates that a 3×3 Conv was used instead of 1×1 Conv for the latent to modulation linear map.