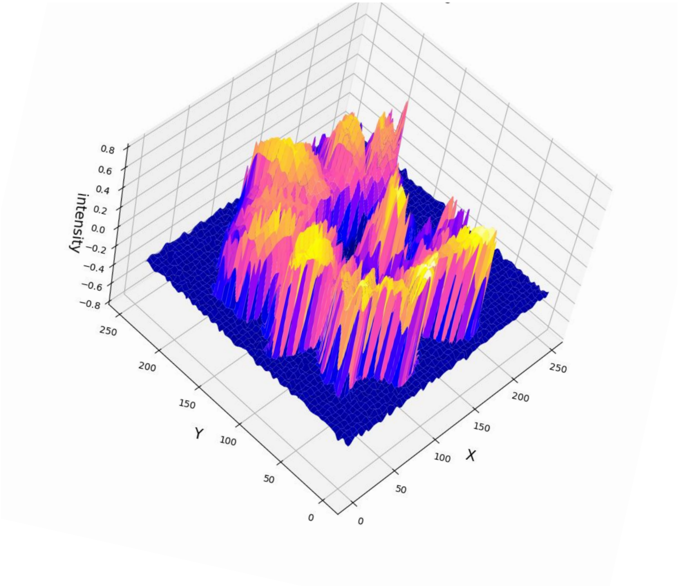
Implicit Neural Representations (INRs)

CS4245 Seminar Computer Vision by Deep Learning

Lecturer: Mahdi Naderibeni
m.naderibeni@tudelft.nl

14 May 2024



TUDelft

Outline

- Physics Informed Machine-Learning/Computer-Vision
- Implicit Neural Representations (INRs)
- Fourier feature maps and Spectral Bias of Neural Networks
- INRs with Periodic Activation Functions
- From Data to Functa!
- Generative AI and INRs
- Editing and Processing INRs

TU Delft

Introduction

Mahdi Naderibeni

PhD student at the pattern recognition lab

Interested in Physics-Informed Machine Learning

Machine Learning for Fluid dynamics; simulating Fluid Flows

Physics and Sora Model

How does Open AI's text-to-video model considers physics in generating videos?

Do the laws of physics hold?

Paper planes
Do some paper planes go missing?
What is happening to the planes at the end of their journey?

Closeup Video of Two Pirate Ships Battling Each Other As They Sail Inside A Cup Of Coffee by Sora

Ref: https://openai.com/index/sora/

Physics and Sora Model

These results are better than what we have ever been capable of.

However, how strong is this argument?

We explore large-scale training of generative models on video data. Specifically, we train text-conditional diffusion models jointly on videos and images of variable durations, resolutions and aspect ratios. We leverage a transformer architecture that operates on spacetime patches of video and image latent codes. Our largest model, Sora, is capable of generating a minute of high fidelity video. Our results suggest that scaling video generation models is a promising path towards building general purpose simulators of the physical world.

Ref: https://openai.com/index/sora/

How to embed physics into Machine Learning

- Augment it in your data, Big corporates' approach.
- Embed it into the model architecture.
- Embed it into the Loss function; Physics-Informed Neural Networks.

What about data itself?

What is the best way of representing signals, images, etc?

Physics is about gradients (rates of change)

Our approach in computing gradients determines our approach in embedding physics into Machine Learning.

$$F = ma$$

$$\mathit{Force} = \mathit{Mass} \cdot \mathit{Acceleration}$$

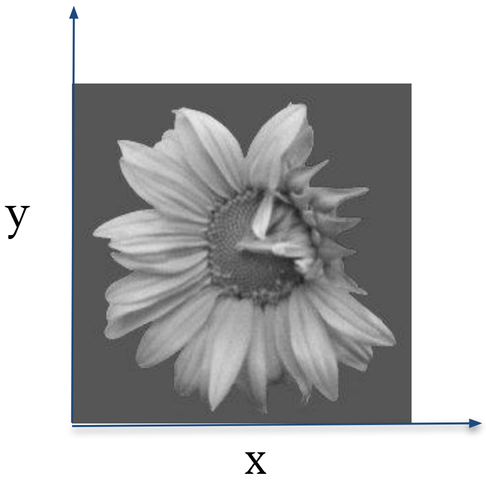
$$\mathit{Acceleration} = \frac{\mathrm{d}(\mathit{velocity})}{\mathrm{d}t}$$

$$\mathit{velocity} = \frac{\mathrm{d}(\mathit{position})}{\mathrm{d}t}$$

TUDelft

Image Gradients

How to compute them?

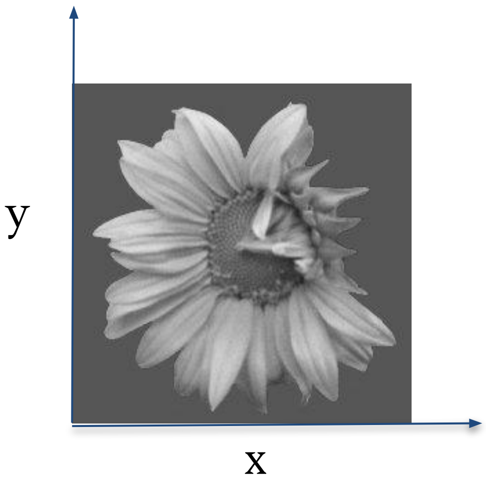


$$f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$$

Image Gradients

How to compute them?

1. Numerical differentiation, (e.g., Finite Difference method)
2. Automatic Differentiation



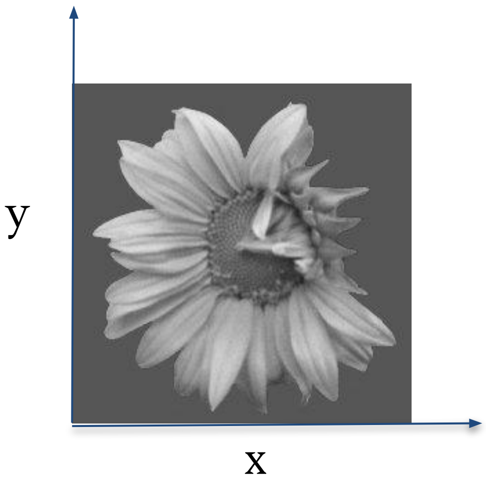
$$f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$$

Image Gradients

How to compute them?

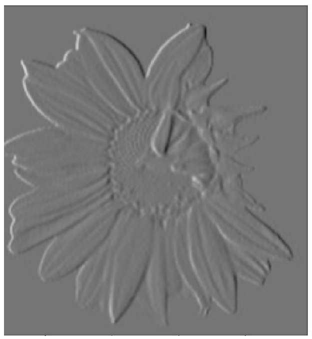
1. Numerical differentiation
2. Automatic Differentiation

$$\nabla f = \left\{ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

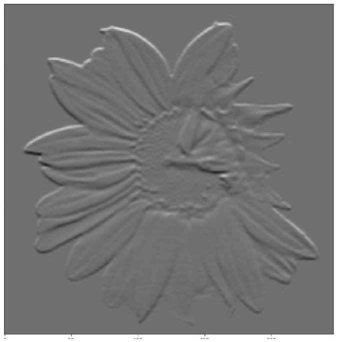


$$f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$$

Gradient Operator



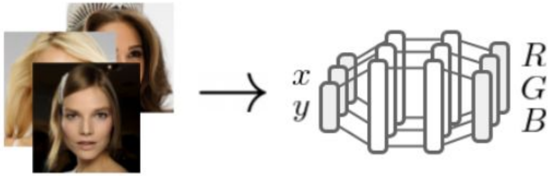
$$\frac{\partial f}{\partial x}$$



$$\frac{\partial f}{\partial y}$$

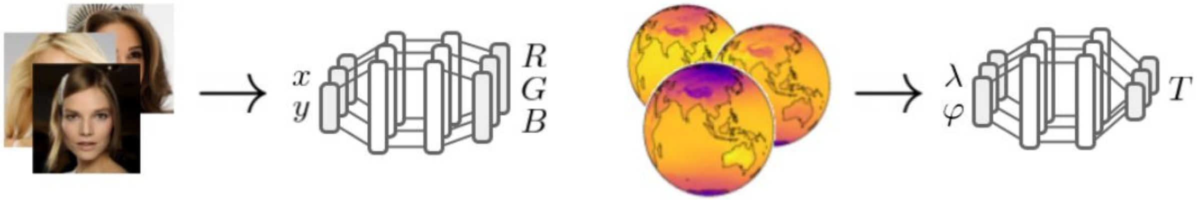
Implicit Neural Representations (INRs)

Represent your data with continuous function



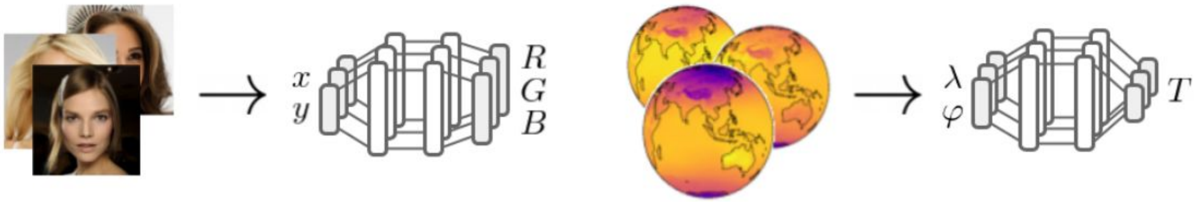
Implicit Neural Representations (INRs)

Represent your data with continuous function



Implicit Neural Representations (INRs)

Represent your data with continuous function

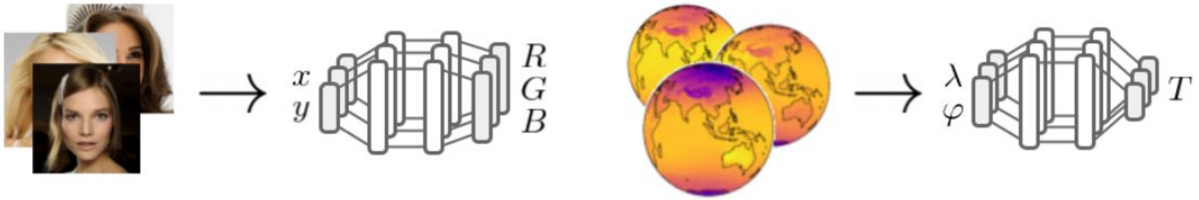


Then what?

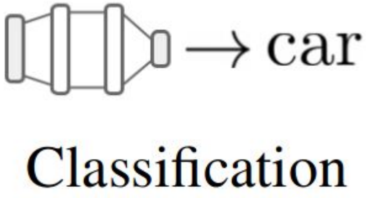
Dupont, E., Kim, H., Eslami, S. M., Rezende, D., & Rosenbaum, D. (2022). From data to functa: Your data point is a function and you can treat it like one. arXiv preprint arXiv:2201.12204.

Implicit Neural Representations (INRs)

Represent your data with continuous function



Use these functions as inputs to your Machine Learning model



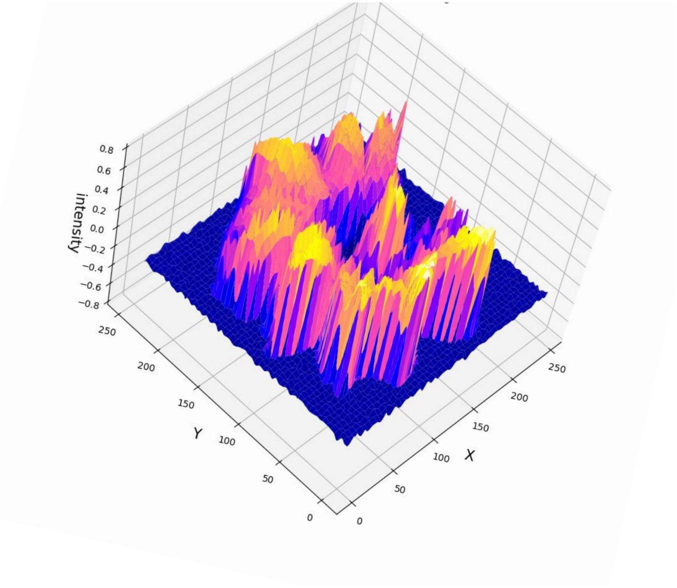
Dupont, E., Kim, H., Eslami, S. M., Rezende, D., & Rosenbaum, D. (2022). From data to functa: Your data point is a function and you can treat it like one. arXiv preprint arXiv:2201.12204.

Implicit Neural Representations (INRs)

Learning an image as a continuous 2D function

f_i :

→ $f_\theta(x_i)$:



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

Implicit Neural Representations (INRs)

Advantages ?

Implicit Neural Representations (INRs)

Advantages ?

1. Agnostic to the sampling resolution

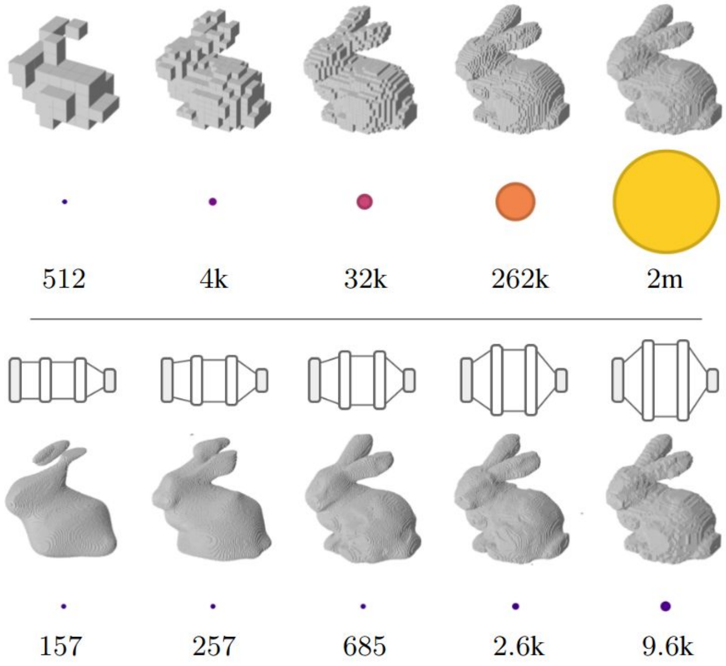
Implicit Neural Representations (INRs)

Advantages ?

1. Agnostic to the sampling resolution

2. Required memory to store INRs is proportional to the complexity of the signal and not to the sampling resolution.

   INRs as a compression algorithm



INRs scale much more gracefully with resolution than array representations. Circle area reflects the numerical size of the array (top) / function (bottom).

TU Delft

Implicit Neural Representations (INRs)

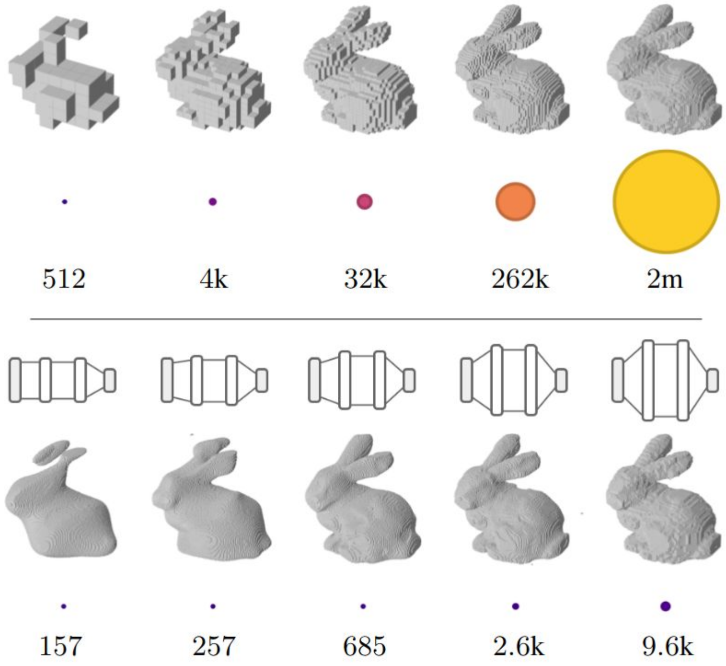
Advantages ?

1. Agnostic to the sampling resolution

2. Required memory to store INRs is proportional to the complexity of the signal and not to the sampling resolution.
   INR as a compression algorithm

3. Use of automatic differentiation to compute input (e.g., image) gradients

4. ?

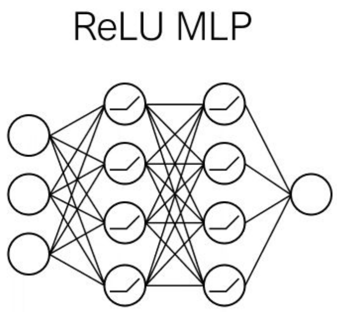


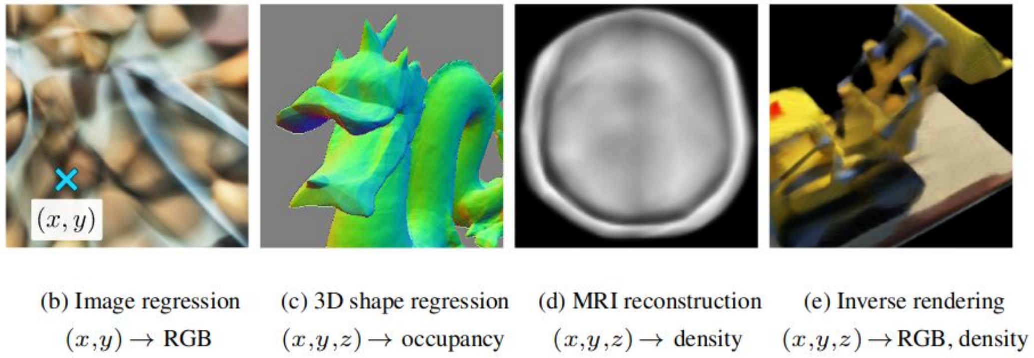
INRs scale much more gracefully with resolution than array representations. Circle area reflects the numerical size of the array (top) / function (bottom).

Dupont, E., Kim, H., Eslami, S. M., Rezende, D., & Rosenbaum, D. (2022). From data to functa: Your data point is a function and you can treat it like one. arXiv preprint arXiv:2201.12204.

Spectral bias in MLPs

MLPs are biased towards learning low frequency functions. They prioritize learning simple patterns that generalize across data samples.

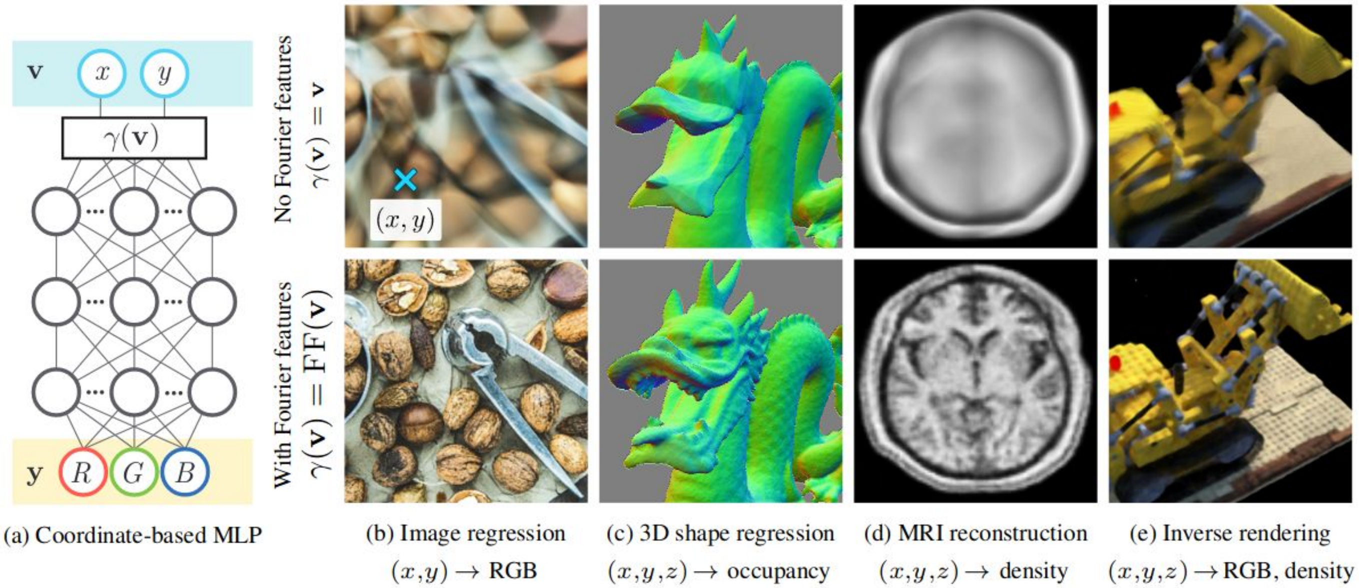




Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., ... & Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, *33*, 7537-7547.

Fourier Feature Maps

- Feature Maps?



Fourier Feature Maps

$$\gamma(\mathbf{v}) = [\cos(2\pi \mathbf{B}\mathbf{v}), \sin(2\pi \mathbf{B}\mathbf{v})]^{\mathrm{T}}$$

γ: feature map , v: input coordinates, B: tunable parameters

- Feature Maps are usually used to transform low dimensional inputs to higher dimensions

- Fourier Feature Maps are helpful to overcome the spectral bias

Why Fourier Feature Maps work?

(a) Coordinate-based MLP (b) Image regression $(x,y) \rightarrow$ RGB (c) 3D shape regression $(x,y,z) \rightarrow$ occupancy (d) MRI reconstruction $(x,y,z) \rightarrow$ density (e) Inverse rendering $(x,y,z) \rightarrow$ RGB, density

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., ... & Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, *33*, 7537-7547.

TU Delft

Fourier Feature Maps

$$\gamma(\mathbf{v}) = \left[\cos(2\pi \mathbf{B}\mathbf{v}), \sin(2\pi \mathbf{B}\mathbf{v})\right]^{\mathrm{T}}$$

γ: feature map , v: input coordinates, B: tunable parameters

- Feature Maps are usually used to transform low dimensional inputs to higher dimensions

- Fourier Feature Maps are helpful to overcome the spectral bias

(a) Coordinate-based MLP

(b) Image regression $(x,y) \rightarrow$ RGB

(c) 3D shape regression $(x,y,z) \rightarrow$ occupancy

(d) MRI reconstruction $(x,y,z) \rightarrow$ density

(e) Inverse rendering $(x,y,z) \rightarrow$ RGB, density

Why Fourier Feature Maps work?

Given that they are effective for the cases of low dimensional data.
They are helpful in decomposing complex signals into several scales of frequencies.

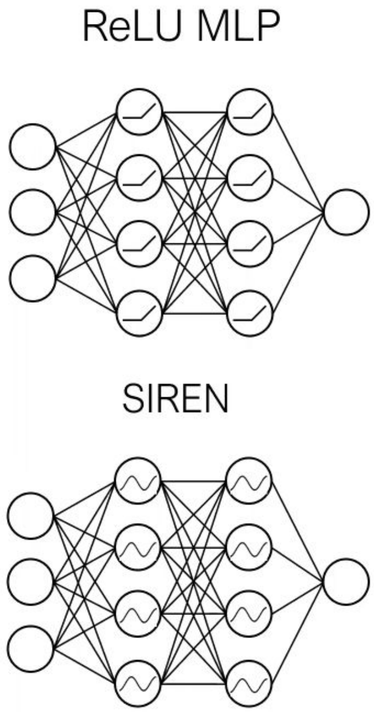
INRs with Periodic Activation Functions

Sinusoidal Representation Networks (SIRENs)

MLPs with sine activation functions outperform ReLU and Tanh MLPs for implicit neural representations.

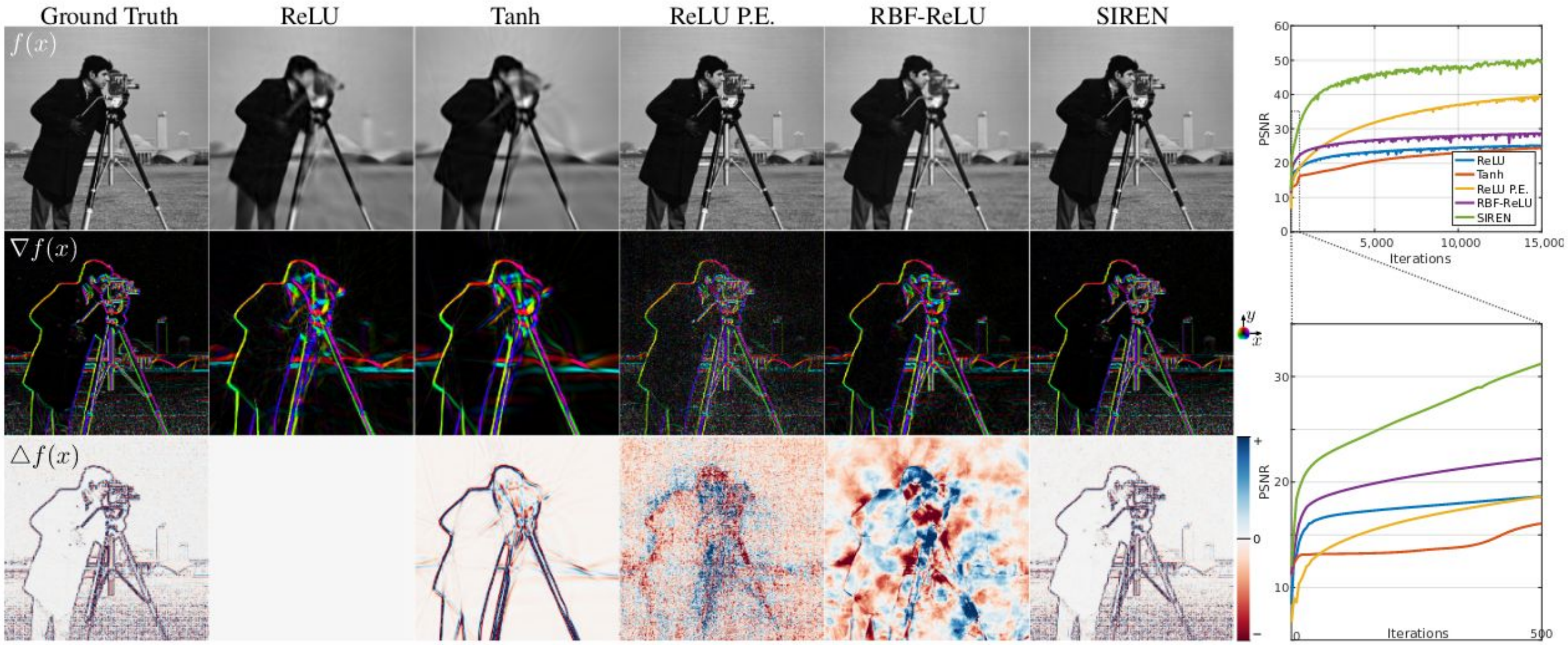
$$y = relu(wx + b)$$

$$y = sin(wx + b)$$



Ref: Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, *33*, 7462-7473.

SIRENs outperform ReLU and Tanh MLPs

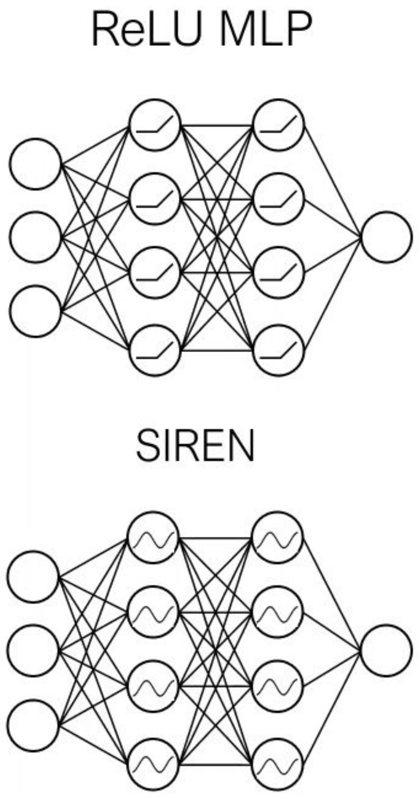


Comparison of different neural network architectures fitting the implicit representation of an image (ground truth: top left). The representation is only supervised on the target image but we also show first- and second-order derivatives of the function fit in rows 2 and 3, respectively.

Ref: Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, *33*, 7462-7473.

INRs with Periodic Activation Functions

Sinusoidal Representation Networks (SIRENs)



Periodic activation functions outperform ReLU and Tanh MLPs for implicit neural representations.

Why SIRENs work and what are the advantages ?

Ref: Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, *33*, 7462-7473.

TU Delft

INRs with Periodic Activation Functions

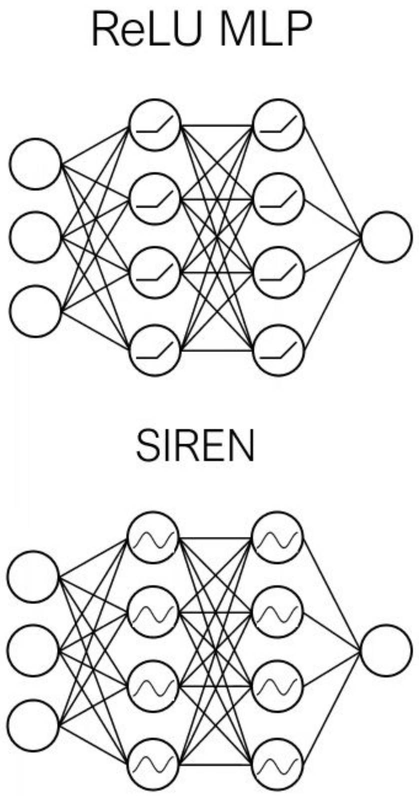
Sinusoidal Representation Networks (SIRENs)

Periodic activation functions outperform ReLU and Tanh MLPs for implicit neural representations.

Why SIRENs work and what are the advantages ?

1. Added complexity
2. Gain some shift-invariance
3. The derivative network has periodic non-linearities

What about the derivative of a ReLU-MLP?

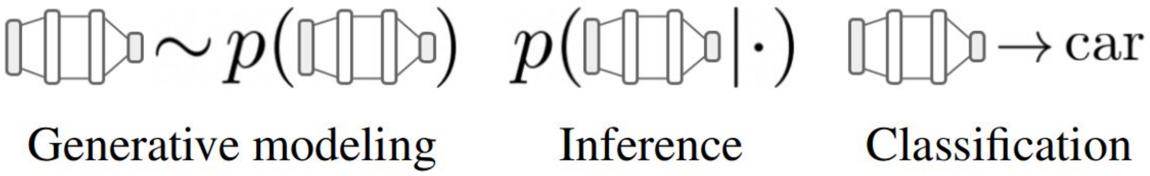


From data to functa

With INRs we have a functional representation of a data sample (e.g., INR's weights).

By training INRs we can transform a dataset to a functional dataset (some call it a **functaset**!)

These functional representation can be used for downstream tasks (e.g., classification, generative modeling).



Deep Generative modelling on functional representations

- Train deep learning directly on the model weights
- Experiments on generative modeling with Normalizing flows, to map a simple base distribution through a sequence of invertible layers parameterized by neural networks.

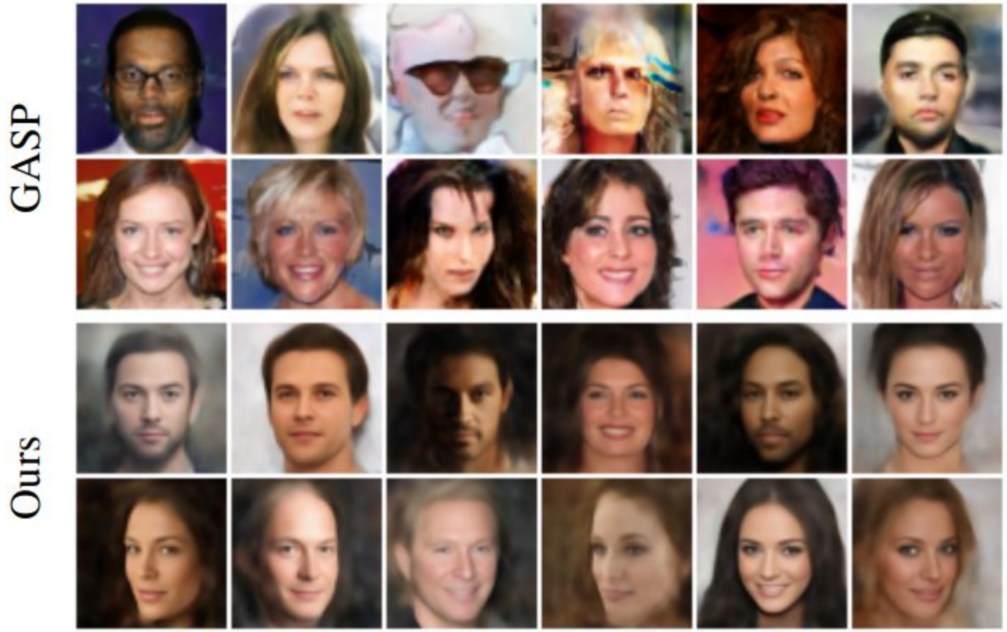


Figure 5. Uncurated samples from GASP and DDPM (diffusion) trained on 256-dim CelebA-HQ 64×64 modulations.

Dupont, E., Kim, H., Eslami, S. M., Rezende, D., & Rosenbaum, D. (2022). From data to functa: Your data point is a function and you can treat it like one. arXiv preprint arXiv:2201.12204.

From data to functional representation

advantages ?

1. Resolution invariance
2. Memory efficiency
3. ?

disadvantages?

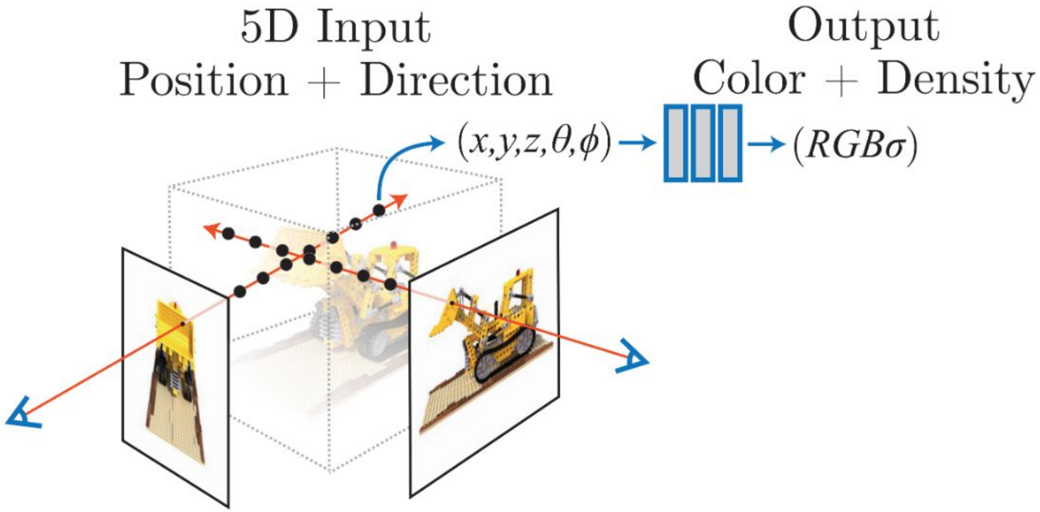
1. Ignoring spatial structure from the data,
2. ?

What else can be done with Implicit Neural Representations?

1. Extend INRs to learn 3D representations from 2D Images
2. ?

Neural Radiance Fields

Reconstructing a three-dimensional representation of a scene from sparse two-dimensional images.



Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, *65*(1), 99-106.

Editing and Processing INRs

How to directly modify an INR without explicit decoding?

Example: Image blending

Gradient-domain image processing

Image 1

$f_1(x)$

Image 2

$f_2(x)$

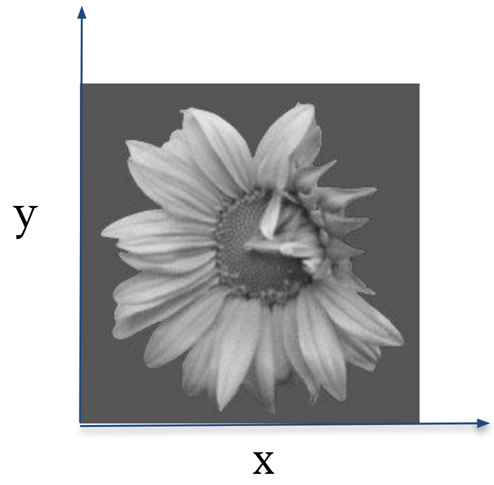
Estimated composite image

$\Phi(x)$

Ref: Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, *33*, 7462-7473.

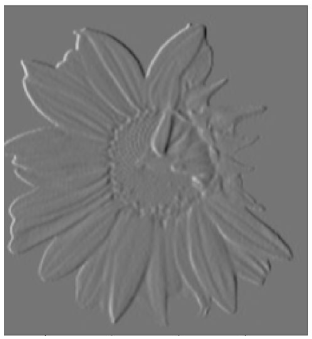
Image Gradients

1. Numerical differentiation, (e.g., Finite Difference method)
2. Automatic Differentiation

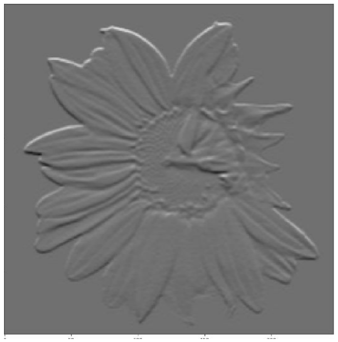


$$f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$$

Gradient Operator



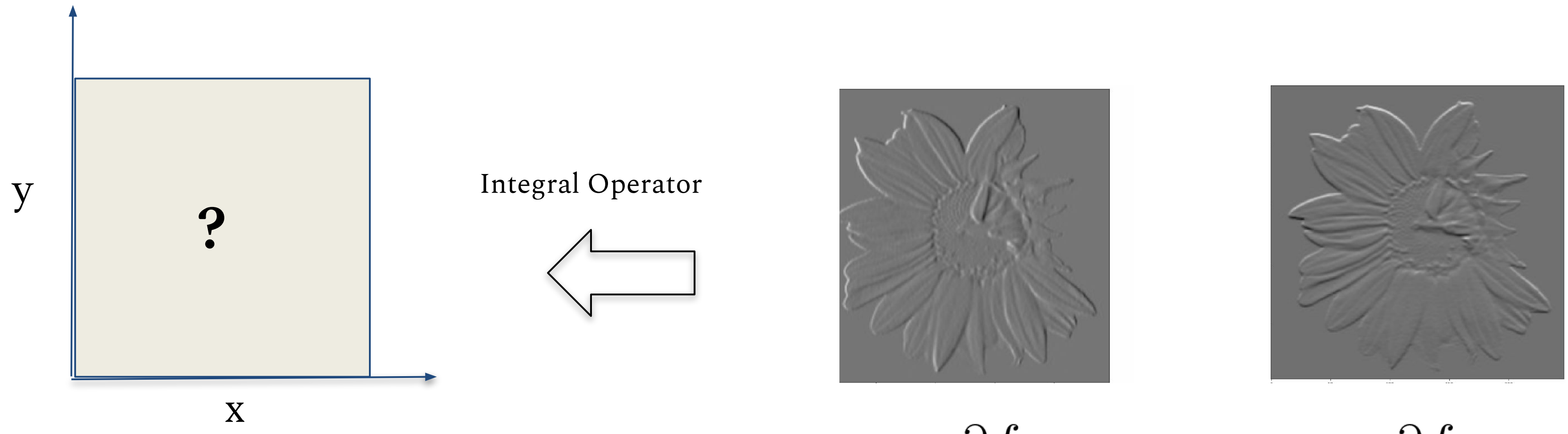
$$\frac{\partial f}{\partial x}$$



$$\frac{\partial f}{\partial y}$$

Integrate image gradients

Can we integrate gradients into a scalar field?

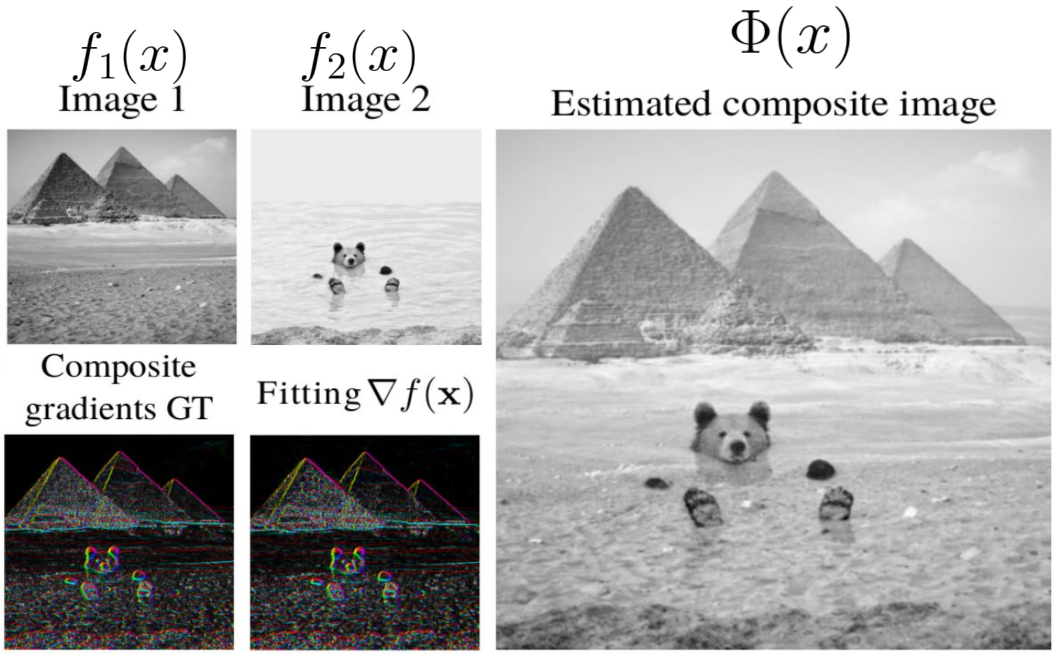


$$f(x, y) : \mathbb{R}^2 \to \mathbb{R}$$

$$\frac{\partial f}{\partial x} \qquad \frac{\partial f}{\partial y}$$

Example: Image blending

Gradient-domain image processing



Loss function:

$$\mathcal{L}_{\text{grad.}} = \int_{\Omega} \|\nabla_{\mathbf{x}} \Phi(\mathbf{x}) - \nabla_{\mathbf{x}} f(\mathbf{x})\| \, d\mathbf{x}$$

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \alpha \cdot \nabla f_1(x) + (1 - \alpha) \cdot \nabla f_2(x), \; \alpha \in [0, 1]$$

Composite gradient

Ref: Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in neural information processing systems, 33*, 7462-7473.

Editing and Processing INRs

How to directly modify an INR without explicit decoding?

To perform tasks like:

(Edge detection, blurring, deblurring, denoising, image inpainting, smoothening, and classification.)

Read more: Signal Processing for Implicit Neural Representations

Ref: Xu, D., Wang, P., Jiang, Y., Fan, Z., & Wang, Z. (2022). Signal processing for implicit neural representations. *Advances in Neural Information Processing Systems*, *35*, 13404-13418.

References

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., ... & Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, *33*, 7537-7547. https://proceedings.neurips.cc/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, *33*, 7462-7473. https://www.vincentsitzmann.com/siren/ https://www.youtube.com/watch?v=Or9J-DCDGko&list=PLat4GgaVK09e7aBNVlZelWWZIUzdq0RQ2&index=4&ab_channel=AndreasGeiger

Dupont, E., Kim, H., Eslami, S. M., Rezende, D., & Rosenbaum, D. (2022). From data to functa: Your data point is a function and you can treat it like one. arXiv preprint arXiv:2201.12204. https://github.com/google-deepmind/functa

Gradient-domain image processing: http://graphics.cs.cmu.edu/courses/15-463/2019_fall/lectures/lecture9.pdf

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, *65*(1), 99-106.

Xu, D., Wang, P., Jiang, Y., Fan, Z., & Wang, Z. (2022). Signal processing for implicit neural representations. *Advances in Neural Information Processing Systems*, *35*, 13404-13418. https://vita-group.github.io/INSP/

Are you interested in Physics-Informed Machine-Learning/Computer-Vision?

Contact me.

Mahdi Naderibeni
m.naderibeni@tudelft.nl