Use INRs as datapoints (functa)

Key concepts of INRs ⇒ INRs : $f_\theta : \mathcal{X} \to \mathcal{F}$ ← features $f \in \mathcal{F}$ (intensity / pixel value)

parameters

coordinates $x \in \mathcal{X}$ (pixel locations)

fitted by minimizing mean squared error over all coordinates locations

$$\boxed{\min_\theta \mathcal{L}(f_\theta, \{x_i, f_i\}_{i \in I}) = \min_\theta \sum_{i \in I} \|f_\theta(x_i) - f_i\|_2^2}$$

each f_θ corresponds to a single image. parameterized by a feedforward neural network (MLP)

"Why functa"

Naive approach for representing functa is to take the parameter vector of SIREN.

↳ Might have large number of parameters X NOT GOOD

↓ So we want ..

Modulations come

uses a shared base network across data points to model common structure, with modulations modeling the variation specific to each data point.

↳ Use modulations instead of parameters - typically much more low dimensional. ⇒ Then we get a latent vector and do downstream work based on it.

KEY IDEA.

So what is modulations?

Usually represented as elementwise affine transformations (shift and scale) applied to the activations of the neural network. In this paper they prove that only use shift can be as good as using both.

Latent Modulated SIREN :

uses a latent modulation vector ϕ that is linearly mapped to the shift modulation.

ith layer is parameterised as:

$$x \mapsto \sin\left(w_0\left(W^{(i)}x + b^{(i)} + s^{(i)}\right)\right)$$

$$s = W'\phi + b'$$

We store this latent vector to represent any given data point.

when creating the functaset we only fit the modulation of each data point with the shared base network fixed.

Inner loop:
update the modulations

Outer loop:
update the base network weights.

Modulation structure: