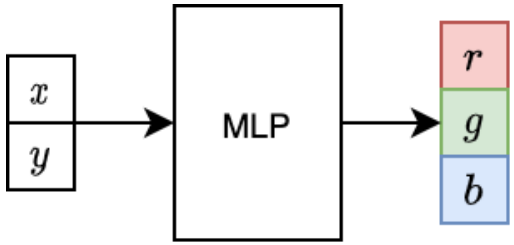
Meta Generation

Ivan Skorokhodov

June 30, 2020

Implicit Neural Representations

- INR is a neural network $f_\varphi(x, y)$ which takes coordinates (x, y) and produces a pixel value:



- We can generate the whole image by computing the value of $f_\varphi(x, y)$ at each coordinate
- I.e. we have *1 image = 1 INR*

Benefits of INRs

- They carry more "complete" information about a signal:
  - Usual 2D arrays of pixels are discretized version of a signal
  - INRs are continuous and carry information "between" the pixels
  - This allows one to have "superresolution out-of-the-box"
- They are popular in 3D deep learning since they are cheaper to generate and operate with than 3D objects
  - For 2D images the situation is not currently clear

Problems with INRs

- INRs are not that small:
  - SIREN/FFN use $\approx 256 \times 5$ MLPs with 3000 epochs
  - But fitting low-frequency images is much easier[1]
- It requires to train an INR on an image to obtain the INR and each training procedure (currently) takes a lot of time (up to 10 minutes).
- Community does not have a good understanding of how to work with them

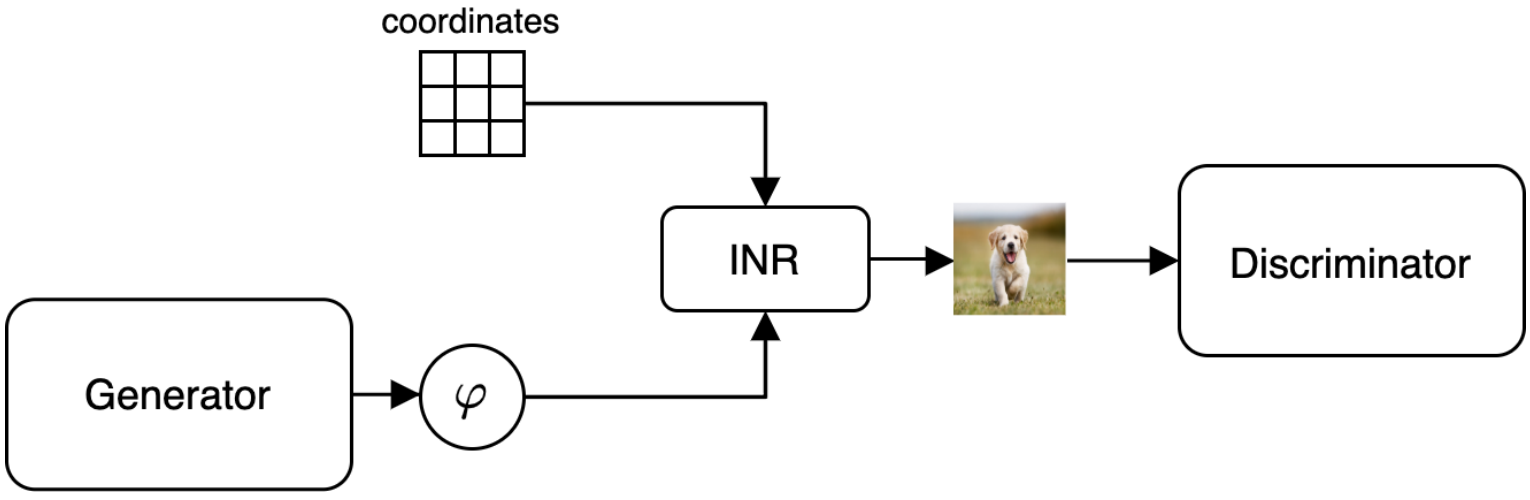
[1] a note to myself: show face/knot examples here

Transposed convolutions are problematic

- They lack the biological plausibility of normal convolutions
- They produce artefacts (like checkerboard patterns)
- You have to change architecture for each output size
  - This can be solved by interpolation procedures to some extent, but for tasks like segmentation this would lead to coarse predictions
- They struggle to capture scene geometry:
  - For example, it's hard to draw long straight lines (so generated walls "wobble", circles are curved, etc)
  - CoordConv paper showed that it cannot learn to classify a pixel given its coordinates

INR-GAN

- Main idea: train a generator to produce INRs



- Discriminator can operate:
  - on top of images
  - on top of INRs (but this would require converting the whole dataset into INRs)

Pros and cons

Pros

- Unified Generator architecture for any domain (images/3D-images/video/audio) since all of them are "representable" by INRs
- Easier super resolution
- Easier progressive growing in Generator
- It should be easer to capture the geometry[2]
- Better biological plausibility

Cons

- I have no idea how hard it is to accomplish
- It can be the case that everyone has the idea of INR-GAN like models (after SIREN and FFNs)
- StyleGAN has a ton of tricks and will be extremely hard to beat

[2] a note to myself: show CoordConv here

Meta Generation

- *A more general idea*:
  - let Decoder take a coordinate + smth (noise? embedding?) and output a pixel value
  - let Generator take smth (noise? embedding?) and produce smth (weights? embedding?) to control[3] Decoder (in the extreme situation, Generator outputs all the Decoder's weights)
- This is pretty similar to StyleGAN
- Ideally, we would like to produce a decoder that can generate a single object (like face identity) in different conditions (lightning, zoom, position, etc)

[3] To "neuromodulate" it — this is a bialogically plausible motivation of hypernetworks

Questions

- How to make INRs cheaper?
- How can we share weights in different decoders (to make INRs cheaper)?
- How can we control Decoder with Generator?